Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Rok Majerčič

Uporaba spletnih virov pri napovedovanju nihanj na trgu digitalnih valut

magistrsko delo

študijski program druge stopnje
računalništvo in informatika

Mentor: doc. dr. Dejan Lavbič

Ljubljana, 2015

Izjava o avtorstvu magistrskega dela

Spodaj podpisani Rok Majerčič sem avtor magistrskega dela z naslovom:

Uporaba spletnih virov pri napovedovanju nihanj na trgu digitalnih valut.

S svojim podpisom zagotavljam, da:

- sem magistrsko delo izdelal samostojno pod mentorstvom doc. dr. Dejana Lavbiča,
- so elektronska oblika magistrskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko magistrskega dela,
- soglašam z javno objavo elektronske oblike magistrskega dela v zbirki "Dela FRI".

V Ljubljani, 11. marca 2015 Podpis avtorja:

Zahvaljujem se mentorju doc. dr. Dejanu Lavbiču za vso pomoč, ki mi jo je nudil v času izdelave te magistrske naloge. Zahvalil bi se tudi sodelavcem, družini ter vsem ostalim, ki so mi v tem času nudili podporo.

Kazalo

Slike

Seznam uporabljenih kratic

kratica	angleško	slovensko
P2P	Peer to Peer	omrežje vsak z vsakim
API	Application Program Interface	programski vmesnik
I/O	Input/Output	vhodno-izhodni
XML	Extensible Markup Language	razširljivi označevalni jezik
RSS	Really Simple Syndication	protokol za objavo spletnih vsebin
BTC	Bitcoin	virtualna valuta bitcoin
GPOMS	Google-Profile of Mood States	Googlovo profiliranje razpoloženjskih stanj
JSON	JavaScript Object Notation	JavaScript objektna notacija
BSON	Binary JSON	binarni JSON
HTML	Hypertext Markup Language	označevalni jezik za oblikovanje večpredstavnostnih dokumentov
CSS	Cascading Style Sheet	prekrivni slogi
VWAP	Volume Weighted Average Price	ovrednotena povprečna cena

Povzetek

Napovedovanje nihanja vrednosti klasičnih finančnih inštrumentov je znan in široko obravnavan problem. Raziskovalci so v preteklosti ta problem reševali s tehnično in temeljno analizo. Prva se pri napovedovanju opira na zgodovinske podatke o valuti oziroma delnici (tržna vrednost delnice oziroma valute ter obseg trgovanja), druga pa analizira zunanje informacije, ki lahko vplivajo na nihanje vrednosti delnice/valute (npr.: predstavitev novega izdelka v podjetju lahko zviša vrednost delnice tega podjetja, medtem ko nižja bonitetna ocena neke države lahko zmanjša vrednost njene valute). V tej magistrski nalogi se bomo osredotočili predvsem na slednjo – temeljno analizo. Uspešnost le-te pa bomo prikazali na dokaj novem, še ne povsem uveljavljenem trgu digitalnih valut. Glavna prednost, ki jo predstavlja trg digitalnih valut v primerjavi s klasičnimi finančnimi trgi, je predvsem njena P2P-narava delovanja. To pomeni, da ima vsak uporabnik vpogled v celotni potek trgovanja (naročila, povpraševanja, ponudbe, transakcije). Poleg tega pa so javno dostopne tudi informacije o delovanju omrežja (porabljena računska moč, količina valute v obtoku, število rudarjev ...).

V prvem, uvodnem delu bo narejen pregled obstoječih metod za napovedovanje nihanja valut, ki jih bomo prevzeli predvsem s področja trgovanja vrednostnih papirjev. V tem delu bomo tudi okvirno predstavili naše izboljšave ter prednosti, ki jih nudi trg digitalnih valut. V nadaljevanju bo podrobno opisano delovanje trga digitalnih valut; opisan bo potek posamezne transakcije, kakšna je vloga rudarjev v omrežju Bitcoin, kako poteka verifikacija uporabnikov in transakcij ter tudi kako je mogoče trgovati z digitalnimi valutami. V naslednjem poglavju bo opisano uporabljeno razvojno okolje (zgradba aplikacije, uporabljena orodja ter knjižnice).

V osrednjem delu bo predstavljen razvoj naše predlagane metode, s katero želimo napovedati nihanje valute Bitcoin. Ta del bo razdeljen na tri osrednje

sklope: podatkovno rudarjenje, analiza zajetih podatkov ter izvedba simulacije. V prvem sklopu bodo opredeljeni spletni viri ter metode, s katerimi se bo izvajal zajem. V drugem sklopu bo izvedena analiza zajetih podatkov, iz katerih bodo izbrani tisti, ki bi lahko vplivali na vrednost valute. V zadnji fazi bodo izbrani podatki uporabljeni v simulaciji napovedovanja nihanja valute, kjer bomo uporabili dva pristopa, večkratno linearno regresijo ter umetno nevronsko mrežo.

Ključne besede: tehnična analiza, temeljna analiza, digitalna valuta, Bitcoin, P2P, napovedovanje, podatkovno rudarjenje, strojno učenje

Abstract

Forecasting volatility of traditional financial instruments is a well known and widely addressed problem. In the past, researches addressed it by using technical and fundamental analysis. The former looks at the past price movement of a currency or a stock (their market value and trading volumes), while the latter analyses outside information which can cause fluctuations in the currency or stock value (e.g.: introducing a new product in the company can increase the value of company's stocks, while lower rating of a country can reduce the value of its currency). The present master's thesis focuses on the latter, i.e. fundamental analysis. Its effectiveness will be demonstrated on a fairly new and not yet well established digital currency market. The main advantage introduced by the digital currency market — in comparison with traditional financial markets — is its P2P nature. This means that every user has an insight into the entire trading process (market orders, demands, offers, transactions). Moreover, the user has access to all the information regarding the functioning of the network (computer power consumption, amount of currency in circulation, numbers of miners...).

The first, introductory part of the thesis offers an overview of existing methods for predicting currency fluctuations that were adopted which mostly come from the field of trading with securities. Moreover, a rough presentation of our improvements and the advantages of the digital currency market are presented. Follows a detailed description on how the digital currency market functions: the process of transactions is described, the role of miners in the Bitcoin Network and the process of verification of users and transactions are explained, and the possibilities of trading with digital currencies are shown. In the next chapter the adopted development environment is described (how the application is built, tools that were used and libraries). The central part of the thesis demonstrates the development of

our proposed method, the goal of which is to predict price movements of Bitcoin. This part is divided into three main parts: data mining, analysis of the considered data and the simulation. In the first part the web resources and methods of data collecting are defined. In the second part, an analysis of the data collected is conducted, on the basis of which only the data that could influence the value of currency is selected. Lastly, the selected data is implemented in a tool simulation which predicts currency fluctuations in which two models were applied: multiple linear regression and artificial neural network.

Key Words: technical analysis, fundamental analysis, digital, Bitcoin currency, P2P, prediction, data mining, machine learning

Poglavje 1

Uvod

1.1 Motivacija

Napovedovanje nihanja vrednosti delnic oziroma valut je problem, s katerim so se investitorji in analitiki v preteklosti velikokrat spopadali. Pri klasičnih finančnih trgih je glavna ovira pomanjkanje prosto dostopnih informacij. Pri delnici nekega podjetja tako lahko trdimo, da je njena vrednost odvisna predvsem od zanimanja vlagateljev po tej delnici (od ponudbe in povpraševanja) ter od velikosti dobička, ki ga to podjetje v nekem trenutku ustvarja. Ti podatki pa na takšnem finančnem trgu niso povsem prosto dostopni, temveč moramo to informacijo razbrati iz ostalih kazalcev (npr. obseg trgovanja, zgodovina nihanja valute, uspešnost poslovanja podjetja v preteklosti – letna poročila, uspešnost poslovanja konkurenčnih delniških družb, naravne nesreče . . .). Povezava teh kazalcev z dejansko vrednostjo delnice je dostikrat nepredvidljiva, zato je nemogoče povsem zagotovo napovedati nihanja v klasičnih finančnih trgih.

Za razliko od klasičnih finančnih trgov pa je vpogled v poslovanje večine trgov digitalnih valut prosto dostopno vsakemu investitorju. Tako so mu na razpolago informacije, kot so:

- celotna zgodovina obsega trgovanja,
- knjiga naročil (obseg povpraševanja in ponudbe),
- celotna zgodovina vseh opravljenih transakcij,

- količina neke valute v obtoku in
- ostali tehnični podatki (npr.: strojna moč, porabljena za izdelavo novih enot valute – hash rate).

V nadaljevanju naloge bomo raziskali, ali lahko z uporabo teh informacij ter z analizo novic in javnega mnenja na socialnih omrežjih točneje napovemo nihanje digitalne valute, kot je to mogoče pri klasičnih finančnih trgih.

1.2 Sorodne raziskave

Digitalne valute in posledično napovedovanje nihanja njihovih vrednosti so še dokaj novo, neraziskano področje, zato bomo izhajali iz raziskav, ki so bile opravljene na področju trga vrednostnih papirjev.

Za namen napovedovanja nihanja vrednosti vrednostnih papirjev so avtorji v [1] analizirali razpoloženje uporabnikov na socialnem omrežju Twitter. V raziskavi so v daljšem časovnem obdobju zajemali objave, ki so se navezovale na opazovano delnico. Pri analizi razpoloženja iz besedila so uporabili dve orodji, OpinionFinder ter GPOMS (Google Profile of Mood States). OpinionFinder določi le polariteto teksta, medtem ko GPOMS razlikuje tekst glede na šest različnih čustvenih stanj (miren, v pripravljenosti, prepričan, ključen, prijazen, vesel). Za časovno vrsto ocene razpoloženj se nato skupaj s časovno vrsto vrednosti delnice izvede Grangerjevo vzorčno analizo [4]. Pri Grangerjevi analizi želimo izvedeti, ali neka časovna vrsta vpliva na neko drugo časovno vrsto v prihodnosti – v opazovanem primeru to predstavlja podobnost polaritete mnenj na Twitterju z vrednostjo delnice v prihodnosti. Trenutni uporabniki digitalnih valut so večinoma tehnološko osveščeni uporabniki, ki za komunikacijo uporabljajo Twitter in ostale sorodne komunikacijske kanale, zato menimo, da bi analiza slednjih tudi v naši raziskavi pripomogla k točnejši napovedi.

Avtorji v [2] za napoved nihanja vrednosti delnice uporabijo umetno nevronsko mrežo. Z modelom, ki ga razvijejo v raziskavi, želijo predvideti, kakšna bo končna vrednost čez pet dni. Model je sestavljen iz petih vhodnih, deset skritih in enega izhodnega nevrona. Vhodni nevroni predstavljajo začetni tečaj, končni tečaj, obseg trgovanja, največjo ter najmanjšo vrednost delnice. V raziskavi so prišli do

zaključka, da število vhodnih nevronov vpliva na točnost predikcije. V ločeni raziskavi [3] se avtorji lotevajo reševanja tega problema z uporabo večkratne linearne regresije. Definirajo model, kjer poleg tipičnih vhodnih podatkov (vrednost delnice) uporablja tudi kazalce, kot so; bruto domači proizvod, indeks cen življenskih potrebščin, cena surove nafte, menjalniški tečaj NOK/USD in NOK/GDP ... V raziskavi želijo ugotoviti, če dejavniki, ki niso neposredno vezani na delnico (npr. cena surove nafte), lahko vplivajo na njeno nihanje. Za omenjeni primer cene nafte se je sicer ugotovila močna korelacijska povezava z vrednostjo delnice, vendar pa jim z razvitim modelom ni uspelo priti do točnejše predikcije.

Tudi na področju napovedovanja nihanja vrednostni digitalnih valut je bilo narejenih že nekaj raziskav. Ena takšnih je objavljena v [10], kjer je za namen trgovalne strategije uporabljena metoda Bayesove regresije. V tem primeru predstavljajo vhodni podatki ponudbe in povpraševanja, ki so zbrani z borze digitalnih valut Okcoin [34]. Uporabi se enostavna trgovalna strategija, kjer je glede na premik vrednosti, določenega z Bayesovo regresijo, simulirana odločitev prodaje oziroma nakupa valute Bitcoin.

1.3 Predlagana rešitev in nadaljnje delo

V našem modelu bomo pri napovedovanju nihanja vrednosti uporabili vhodne podatke, ki jih bomo pridobili iz več različnih virov (knjiga naročil, zgodovina trgovanja, objave na omrežju Twitter, novice, statistični podatki valute Bitcoin ...). Omenjene podatke bo treba v daljšem časovnem obdobju predhodno zbirati in jih shranjevati v nerelacijsko podatkovno bazo. Ko bo ta množica podatkov dovolj velika, bomo lahko pričeli z izdelavo samega modela. Informacije, koristne za našo raziskavo, bomo pridobili z naslednjimi načini procesiranja podatkov:

- normalizacijo,
- ugotavljanjem zakonitosti iz teksta,
- združevanjem podatkov in
- interpolacijo časovne vrste.

S tako pripravljenimi podatki bomo nato poskusili poiskati njihovo povezavo z vrednostjo valute. To bomo storili z izračunom korelacijskega faktorja med vrednostjo valute ter časovnim zamikom različnih vhodnih parametrov. Glede na dobljene rezultate bomo nato zgradili modela, kjer bomo uporabili dve metodi – umetno nevronsko mrežo ter večkratno linearno regresijo.

Metodi bomo razdelili na dve fazi, fazo učenja modela ter fazo simulacije. V vsaki fazi bomo uporabili tudi ločeno množico podatkov. S tem pa bomo pokazali robustnost modela, da je mogoče z vhodnimi podatki, ki niso bili uporabljeni pri učenju modela, predvideti nihanje valute.

Poglavje 2

Trg digitalnih valut

2.1 Digitalna valuta Bitcoin

Bitcoin [6] je v času pisanja te naloge najbolj razširjena digitalna valuta, zato se bomo v nadaljevanju osredotočili nanjo. V obtok je bil predstavljen leta 2009, sicer pa je decentralizirana digitalna valuta, ki temelji na P2P-omrežju. To pomeni, da uporabniki lahko trgujejo med seboj brez centralne entitete, kot je banka pri klasičnih finančnih trgih.

2.2 Opis omrežja Bitcoin

Na Sliki 2.1 je prikazana osnovna shema omrežja Bitcoin. Entiteti, ki trgujeta v omrežju, sta označeni kot pošiljatelj (sender) in prejemnik (receiver). Ti entiteti imata na svoji napravi (PC, tablica, mobilni telefon ...) nameščeno aplikacijo, ki jo imenujemo "denarnica".

Vsaka denarnica vsebuje poljubno število Bitcoin naslovov. Bitcoin naslov vsebuje od 26 do 35 alfa numeričnih znakov (npr.: 3J98t1WpEZ73CNmQviec rnyiWrnqRhWNLy), predstavljamo pa si ga lahko kot številko bančnega računa pri klasičnih finančnih inštrumentih. Vsak takšen naslov ima nase vezano tudi stanje oziroma bilanco (število Bitcoinov, s katerimi lahko razpolagamo). Za razliko od klasičnega finančnega računa pa lahko v omrežju Bitcoin uporabnik kreira praktično neomejeno število računov – priporočljivo je celo za vsako transakcijo

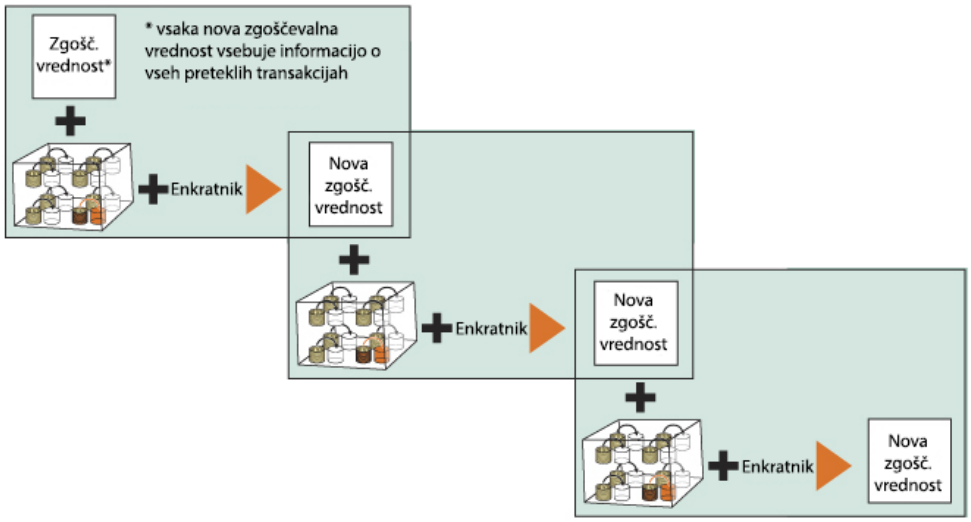
Slika 2.1: Bitcoin omrežje

kreirati nov račun.

Transakcija med dvema entitetama (npr.: če želi Alice nazati 1 Bitcoin na Bobov račun) se izvede tako, da najprej obe entiteti kreirata nov naslov v svoji denarnici. S kreiranjem naslova klient (denarnica) ustvari tudi par ključev (privatnega in javnega). Njun namen je verificiranje entitet, ki so prisotne pri transakciji. Prenos valute se izvede tako, da Alice svojo transakcijo podpiše s svojim privatnim ključem ter tako šifrirano sporočilo pošlje Bobu. Ta lahko nato to sporočilo verificira z uporabo Alicinega javnega ključa, ki je javno viden, s tem pa Bob verificira, da transakcija res prihaja od Alice.

Verifikacija transakcij se izvaja z uporabo rudarjev. Njihova naloga je, da s svojo strojno opremo (PC, ASIC . . .) računajo kriptografske zgoščevalne funkcije. Namen zgoščevalne funkcije pa je, da vhodne nize različnih dolžin pretvori v izhodne nize enakih dolžin. Vsaka najmanjša sprememba na vhodnem nizu vpliva na izhodni niz, zaradi česar je nemogoče predvideti, kakšen izhodni niz bo kreiral nek vhodni niz.

Glavna značilnost valute Bitcoin je, da je vsa zgodovina transakcij zapisana v t. i. bločni verigi, katere kopijo ima vsak uporabnik zapisano v svoji denarnici. Bločna veriga predstavlja zaporedje zgoščevalnih vrednosti (Slika 2.2). Novo



Slika 2.2: Bločna veriga

zgoščevalno vrednost pa rudarji izračunajo okvirno vsakih 10 minut, s čimer verificirajo vse transakcije, ki so se izvedle v tem časovnem obdobju. Novo zgoščevalno vrednost se izračuna iz:

- zgoščevalne vrednosti, ki vsebuje vse pretekle transakcije,
- bloka, ki vsebuje vse transakcije, ki so se izvedle v zadnjih 10 minutah (transakcije, ki niso verificirane) in
- enkratnika.

Enkratnik je naključno število, njegova naloga pa je, da je vsaka nova zgoščevalna vrednost v verigi unikatna (kljub enakim vhodnim podatkom).

Naloga rudarjev pa je ravno "ugibanje" omenjenih enkratnikov. Z računanjem zgoščevalne funkcije, kjer združijo preteklo zgoščevalno vrednost, neverificirane transakcije ter naključni enkratnik, želijo na izhodu dobiti vrednost, ki mora biti v točno določeni obliki (zaporedje ničel na začetku niza). Ko rudar poišče pravilen enkratnik (ter posledično zgoščevalno vrednost), se kreira nov blok, ki se doda na konec bločne verige. Rudar je za svoje delo poplačan s t. i. nagrado, ki se nakaže

na njegovo denarnico. Sistem z uporabo nagrad skrbi, da je v omrežju vedno dovolj rudarjev, ki skrbijo za verificiranje novih transakcij.

2.3 Borze za trgovanje z digitalnimi valutami

Borzo z digitalnimi valutami lahko primerjamo s tradicionalno borzo vrednostnih papirjev. Je mesto (oziroma v našem primeru spletna platforma), kjer se na enem mestu srečajo ponudniki ter povpraševalci, ki želijo trgovati z določeno digitalno valuto. Trenutne večje aktivne borze so:

- OKCoin: https://www.okcoin.com/,
- BitStamp: https://www.bitstamp.net/,
- BTCe: https://btc-e.com/ in
- Bitfinex: https://www.bitfinex.com/.

Vsakdo, ki želi trgovati z določeno digitalno valuto, mora na takšni borzi registrirati račun, na katerega prenese finančna sredstva v klasični (USD, EUR ...) ali digitalni (Bitcoin, Litecoin ...) valuti. Uporabnik lahko nato trguje z ostalimi trgovci, kjer s svojimi sredstvi kupuje ostale valute. Borza digitalnih valut večinoma podpira različne trgovalne pare in s tem omogoča trgovanje med različnimi digitalnimi in klasičnimi valutami (BTC/USD, LTC/USD, BTC/LTC ...). Nakup oziroma prodaja digitalnih valut se izvaja preko spletnega grafičnega vmesnika oziroma programskega vmesnika (API-ja), s katerim lahko postopek trgovanja avtomatiziramo.

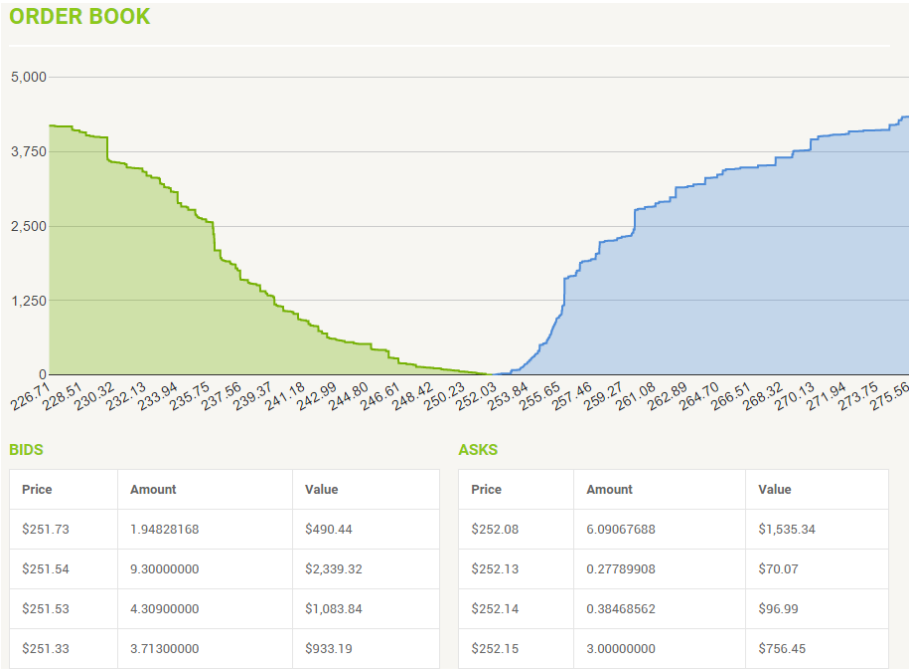
2.3.1 Postopek trgovanja

Pri trgovanju z digitalnimi valutami so prisotne tri osrednje entitete − prodajalec, kupec in posrednik (borza):

- kupec poda količino ter vrednost, po kateri bi bil pripravljen kupiti neko valuto;
- prodajalec poda količino in vrednost, po kateri bi bil pripravljen prodati neko valuto;

- borza združi kupca in prodajalca z enako ponudbo/povpraševanjem.

Začetek trgovanja se torej prične tako, da kupci in prodajalci podajo svoje ponudbe in povpraševanja, ki so zapisana v t. i. knjigo naročil (Slika 2.3). Obseg naročil ter povpraševanj si najlažje vizualiziramo z uporabo grafa, ki prikazuje globino trga (Slika 2.3). Na x osi so zapisane vrednosti valute, po kateri se ta kupuje oziroma prodaja; na y osi pa je prikazan obseg ponudb/povpraševanj pri določeni vrednosti valute. Levi graf prikazuje obseg ponudb in pričakovano se ta niža z večanjem vrednosti valute, saj želi kupec kupiti neko valuto po čim manjši vrednosti. Desni graf prikazuje povpraševanja in za razliko od ponudb se obseg slednjega veča z vrednostjo valute – prodajalec želi iztržiti čim večji znesek. Kjer se stikata levi in desni graf, je dejanska vrednost valute, s katero se entiteti (kupec in prodajalec) strinjata, zaradi česar se bo izvedla transakcija. S tem je transakcija med kupcem in prodajalcem zaključena ter se zabeleži v seznam zgodovine trgovanj (Slika 2.4). Zgodovina trgovanj (obseg trgovanja, začetna, končna, najvišja in najnižja vrednost valute) predstavlja ključni kazalec pri tehnični analizi nihanja valute. Vse omenjene vrednosti lahko združimo in prikazujemo na enem samem grafu – z uporabo japonskih svečnikov (Slika 2.5).



BIDS

Price	Amount	Value
$251.73	1.94828168	$490.44
$251.54	9.30000000	$2,339.32
$251.53	4.30900000	$1,083.84
$251.33	3.71300000	$933.19

ASKS

Price	Amount	Value
$252.08	6.09067688	$1,535.34
$252.13	0.27789908	$70.07
$252.14	0.38468562	$96.99
$252.15	3.00000000	$756.45

Slika 2.3: Primer knjige naročil borze Bitstamp

LIVE TRADES

Time since	Amount	Price
1 minutes	0.24024001 BTC	$252.69
1 minutes	0.21646831 BTC	$252.68
3 minutes	0.10980000 BTC	$252.20
3 minutes	0.05024977 BTC	$252.69
4 minutes	0.38858804 BTC	$252.69

Slika 2.4: Seznam trgovanj

Slika 2.5: Prikaz zgodovine trgovanja z japonskimi svečniki

Poglavje 3

Predlagani inteligentni sistem za zajem, analizo ter simulacijo trgovanja

3.1 Zgradba inteligentnega sistema

Na sliki 3.1 si lahko ogledamo shemo uporabljenega inteligentnega sistema. Njihova logika je razdeljena na dva osrednja sklopa, pregledovalnike in poti (routes). Pregledovalnik vsebuje spodaj opisane module module, kjer vsak komunicira z svojim spletnim virom preko API-ja:

- Exchange_reader – komunicira s spletno borzo Bitstamp [33], ki trguje le z digitalno valuto Bitcoin. Borzo Bitstamp smo izbrali zaradi njene dolge prisotnosti na trgu ter njenega velikega obsega trgovanja (v obsegu trgovanja ga prehitita le BTCChina [31] in Bitfinex [32]). Vrednost valute se po različnih borzah bistveno ne razlikuje, zato smo se odločili, da bomo spremljali le eno samo borzo digitalnih valut;

- Tweet_reader – komunicira s pretočnim Twitterjevim API-jem. Z modulom je mogoče spremljati objave, ki se nanašajo na definirano ključno besedo. Te objave pa je mogoče dodatno filtrirati glede na njihove lastnosti (pomembnost, jezik objave, število sledilcev ...);

- Bcinfo_reader – periodično shranjuje statistične podatke o omrežju Bitcoin iz spletnega vira blockchain.info [35];
- Feed_reader – spremlja spletne novičarske portale, ki objavljajo novice o valuti Bitcoin.

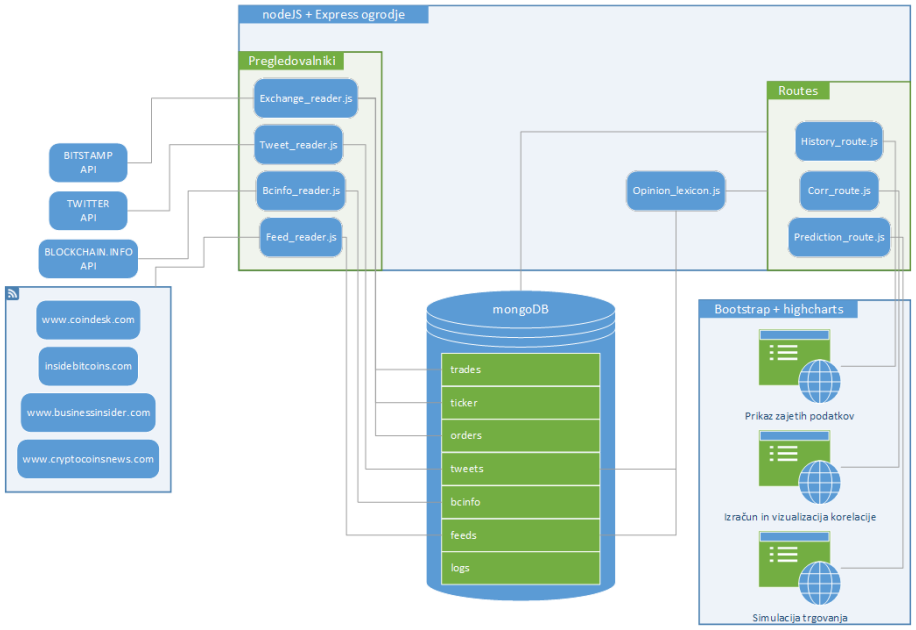
Podatki, ki jih pregledovalniki zajamejo, so nato shranjeni v podatkovno bazo, z njimi pa nato lahko razpolagamo v treh različnih modulih, ki se nahajajo v mapi "routes":

- History_route – skrbi za enostaven grafični prikaz zajetih podatkov;
- Corr_route – izračun podobnosti vrednosti valute Bitcoin z različnimi časovnimi vrstami (spletnimi viri). Tu smo se omejili na izračun križne korelacije ter T-testa;
- Prediction_route – vsebuje predlagan model za napovedovanje vrednosti valut z možnostjo simulacije trgovanja. Omejili se bomo le na zajete podatke, ki so se izkazali, da imajo veliko stopnjo podobnosti z vrednostjo valute.

Jedro sistema predstavlja strežniško okolje node.js v kombinaciji z nerelacijsko podatkovno bazo mongoDB. Node.js okolje smo izbrali, ker uporablja dogodkovno arhitekturo ter neblokirajoč I/O API.

Pri dogodkovni arhitekturi so zahtevki na oddaljen strežnik poslani glede na spremembo določenega stanja. Spremembo stanja preverja dogodkovni emitor, ki ga v okolju node.js imenujemo tok podatkov (stream). Če predstavimo dogodkovno arhitekturo na primeru spremljanja objav na omrežju Twitter, bo tok podatkov spremljal spremembo stanj. Ko tok podatkov zasledi spremembo stanja, ki v našem primeru predstavlja novo objavljeno sporočilo, ta aktivira povratni klic (callback). Ta z drugimi besedami predstavlja funkcijo, ki se pokliče po vsaki spremembi stanja, ta pa skrbi za nadaljnjo obdelavo sporočila (npr: shranjevanje v podatkovno bazo).

Pri neblokirajočem vhodno-izhodnem API-ju so vsi klici izvedeni asinhrono. To pomeni, da sistem ne čaka na odgovor nekega zahtevka, temveč se koda neprekinjeno izvaja naprej. Če tako nek dogodek aktivira povratni klic ter pokliče neko funkcijo, ki se že izvaja, se bo funkcija za ta dogodek začela izvajati brez čakanja. Prednost take arhitekture je predvsem v hitrosti, saj je koda napisana



Slika 3.1: Shema inteligentnega sistema

tako, da ne prihaja do nepotrebnih prekinitev. To pa je bistvenega pomena pri zajemu spletnih virov, saj lahko v kratkem časovnem obdobju prejmemo veliko število zahtev, ki jih je potrebno obdelati sočasno.

Podatkovno bazo mongoDB smo izbrali zaradi nerelacijske arhitekture – ni potrebno predhodno definiranje zgradbe podatkovne baze. Ker se lahko odgovori iz določenega strežnika s časom spreminjajo (npr.: Twitter doda novo kategorijo v objavljenem sporočilu), to predstavlja ključno prednost pri zajemu spletnih virov, saj ni potrebno sprotno prilagajanje baze. Druga lastnost, zaradi katere smo izbrali omenjeno bazo, je njen format zapisa BSON, ki je nadgradnja zapisa JSON. Ker bodo vsi odgovori, ki jih bomo sprejeli iz zunanjih API-jev (Twitter, Bitstamp, BlockChain.info . . .) zapisani v JSON-formatu, bo to močno olajšalo shranjevanje sprejetih podatkov v podatkovno bazo, saj ne bo potrebna predhodna obdelava le-teh.

Naš sistem bo vseboval tudi uporabniški vmesnik, na katerem bodo predstavljeni rezultati, zato smo zaradi lažjega razvoja izbrali spletno ogrodje Express.js [22]. To ogrodje poveže podatke, shranjene v bazi, z zaledjem (backend) ter

uporabniškim vmesnikom (frontend). V datoteki app.js, ki se nahaja korenskem imeniku naše aplikacije, tako lahko definiramo funkcijo, ki se zažene v primeru določenega spletnega zahtevka. V spodnjem primeru se ob obisku spletnega mesta "/draw_graph" zažene funkcija "drawGraph", ki se nahaja v datoteki /routes/history.js.

```
var routesHistory = require('./routes/history');
app.get('/draw_graph', routesHistory.drawGraph);
```

Osrednja programska logika je izvedena v datotekah, ki se nahajajo v mapi "routes". Tu se obdelujejo poizvedbe, pošiljajo zahtevki na podatkovno bazo ter kličejo spletne grafične predloge, kamor so poslani obdelani podatki, kateri bodo vidni uporabniku preko spletnega brskalnika.

Grafične predloge bomo kreirali v grafičnem pogonu jade. Nivo html značk je v tem pogonu definiran z uporabo zamikov, prav tako pa ni treba zapisovati zaključnih značk. V grafičnem vmesniku smo uporabili še HTML/CSS ogrodje Bootstrap [23] ter knjižnico highcharts [24]. Knjižnica Highcharts omogoča izdelavo različnih tipov interaktivnih grafov in diagramov.

3.2 Opis procesa

Celoten proces napovedovanja nihanja valute bo razdeljen v tri faze:

- zajem podatkov iz spletnih virov,
- analizo zajetih podatkov in
- simulacijo trgovanja.

V prvi fazi bomo z uporabo pregledovalnikov iz različnih spletnih virov preko programskih vmesnikov zajemali podatke v JSON-formatu. Večinoma bo šlo tu za tokovni način komunikacije, kjer bodo zahtevki poslani glede na dogodek – nova objava na oddaljenem strežniku. Pri nekaterih virih pa bomo namesto tokovnega načina pošiljali enostavne HTTP-zahtevke, ki bodo periodično poslani na strežnik (npr.: spletni vir BlockChain.info). JSON-odgovor bo potem shranjen (neposredno ali z predhodnim procesiranjem podatkov) v pripadajočo zbirko v podatkovni bazi mongoDB.

V drugi fazi bo narejena analiza sprejetih podatkov. Njena logika slednje bo zapisana v datoteki "corr_route.js". Namen te faze je analiza zajetih podatkov ter ugotovitev primernosti le-teh za uporabo pri napovedovanju nihanja valute. Tu bomo z asinhronimi klici (knjižnica Async.js [25]) sočasno naredili poizvedbo iz dveh podatkovnih baz (zbirka "ticker", ki vsebuje vrednost valute, ter vsakega od spletnih virov). Te bomo nato primerjali med seboj z uporabo križne korelacije. Ta bo določila kazalec, ki predstavlja razmerje med vrednostjo valute ter različnimi spletnimi viri.

V zadnji fazi bomo primerne spletne vire, ki smo jih izluščili v prejšnji fazi, uporabili pri napovedovanju valute. Njena logika se nahaja v datoteki "prediction_route.js". Tu bosta uporabljena dva učeča se sistema: linearna regresija ter umetna nevronska mreža. Oba delujeta v dveh zaporednih fazah, prva je faza učenja, druga pa faza simulacije. V fazi učenja so znani vhodi in izhodi sistema. Z vhodi in izhodi nato podajamo sistem ter ga naučimo, kako naj se odziva na določene vhode. V fazi simulacije uporabimo ločen niz podatkov od tistega, ki smo ga uporabili v fazi učenja. Bistvo te faze je, da podajamo sistem samo z vhodnimi podatki, ta pa glede na testne podatke (ki jih je sprejel v fazi učenja) aproksimira približek izhoda.

Poglavje 4

Zajem podatkov iz spletnih virov

Preden začnemo z analizo trga digitalnih valut ter z izdelavo trgovalnega modela je potrebno s spleta zbrati informacije, ki se nanašajo na opazovano digitalno valuto. Razvili smo servis, ki v realnem času spremlja različne vire ter periodično shranjuje vse na novo objavljene informacije.

4.1 Pregled spletnih virov

V nadaljevanju bomo naredili kratek pregled vseh spletnih virov, iz katerih smo zajemali podatke. Tu bodo zajete borze digitalnih valut, socialno omrežje, novičarski portali in tudi spletna storitev, ki vodi statistiko o delovanju Bitcoin omrežja.

4.1.1 Twitter

Twitter je socialno omrežje, ki uporabnikom omogoča objavo kratkih sporočil, dolgih do 140 znakov. Z njimi lahko uporabnik hitro in ažurno posodablja svoj status z nekaj besedami, v katerih zajame bistvo. Zaradi enostavnosti sporočil se javno mnenje zelo hitro širi po omrežju, saj uporabniki porabijo relativno malo časa za objavo svojega sporočila. Večina objav je tudi javno dostopnih, kar pomeni, da so te vidne neregistriranim uporabnikom ter uporabnikom, ki ne sledijo določeni osebi. Zaradi teh značilnosti (enostavnost, ažurnost ter javna dostopnost) Twiter

predstavlja odlično statistično orodje za analizo javnega mnenja o nekem opazovanem področju – v našem primeru nas zanima javno mnenje o digitalni valuti Bitcoin.

Iz uporabniškega vidika je sporočilo na Twitterju sestavljeno iz imena pošiljatelja (uporabnikovega imena s predpono @) ter jedra sporočila. Jedro sporočila je dolgo do 140 znakov in poleg besedila vsebuje tudi označbe (hashtags) ter povezave. Označba je določena beseda v objavi s predpono #, z njo pa označimo področja, na katera se sporočilo nanaša.

Je pa uporabniku viden le del celotnega sporočila. Če si ogledamo JSON-odgovor sporočila (Slika 4.1), lahko vidimo, da sporočilo vsebuje še kup dodatnih informacij. Vsako sporočilo tako vsebuje identifikator, tekst sporočila, podrobne informacije o avtorju sporočila, informacije o ponovni objavi (retweet), zgradbi sporočila, pomembnosti sporočila, kraju in času objave ... O uporabniku so na voljo informacije, kot so: pravo ime, lokacija, časovni pas, število sledilcev ... Vsi ti parametri omogočajo analitiku dodatno analizo in filtriranje zajetih sporočil.

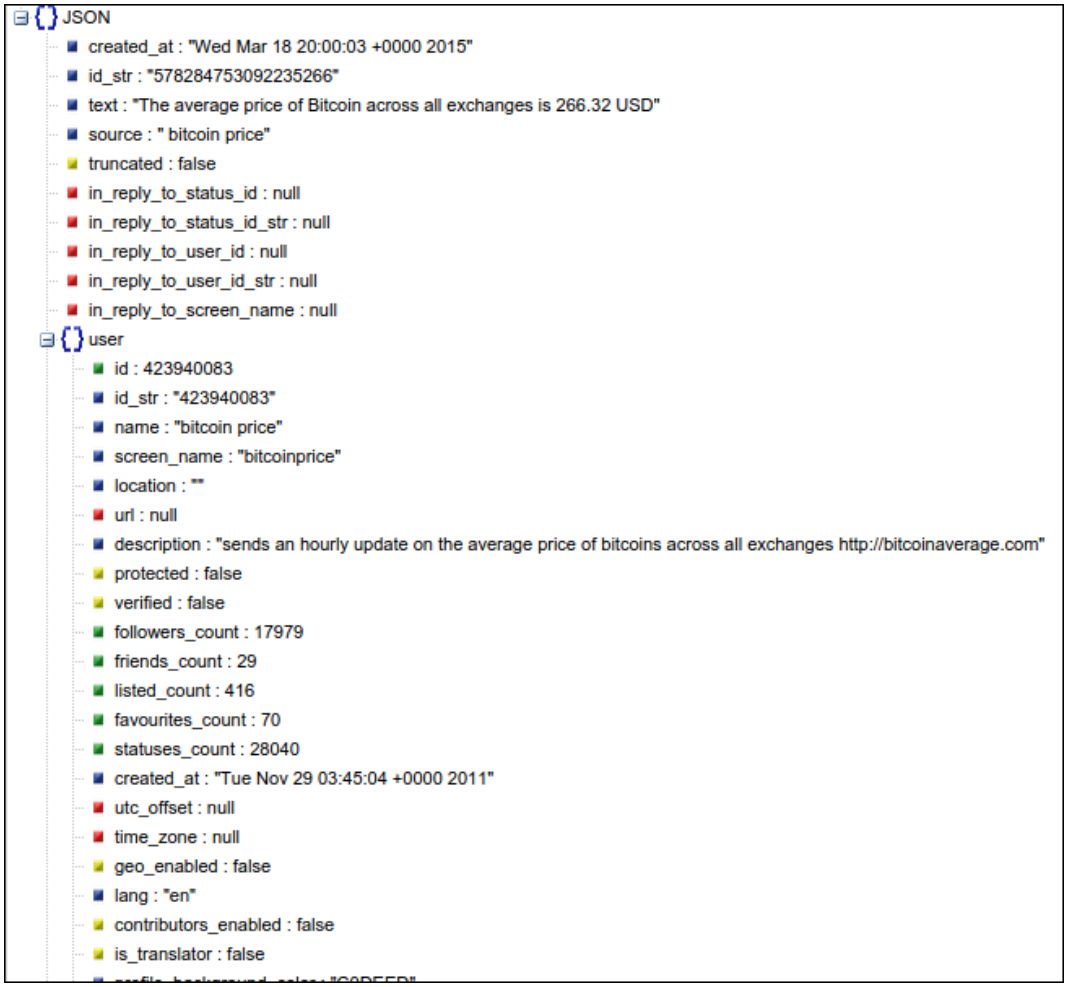
Za spremljanje in analizo sporočil se najbolj pogosto uporablja uradni Twitterjev pretočni programski vmesnik, ki sprejema zahtevke preko POST-protokola. Pretočni programski vmesnik deluje tako, da na Twitterjev strežnik pošljemo HTTP-zahtevo, v odgovor pa sprejmemo podatkovni tok. Ta tok ostane odprt, dokler ga odjemalec ne prekine. V času, ko je podatkovni tok vzpostavljen, se ažurno spremlja opazovane uporabnike oziroma predhodno filtrirana področja.

Twitter podpira tri različne programske vmesnike, ki uporabljajo pretočni način zajema podatkov:

- javni tok – spremljanje določene tematike,
- uporabniški tok – spremljanje posameznega uporabnika in
- spletni tok – spremljanje skupine uporabnikov.

Za naš namen je najprimernejši javni tok, saj tako spremljamo tok vseh javno objavljenih sporočil. Slednjega se nadalje deli na tri različne programske vmesnike:

- POST statuses/filter – vrne objave, ki ustrezajo definiranim parametrom,
- GET statuses/sample – vrne naključni vzorec objav in



Slika 4.1: Izsek Twitter sporočila v JSON-formatu

- GET statuses/firehose – vrne vse nove objave (brez filtriranja).

Spremljati želimo le objave, ki se navezujejo na digitalne valute, zato je za nas primeren programski vmesnik "POST statuses/filter". Objavam lahko sledimo po treh glavnih parametrih: avtorju, ključnih besedah ter lokaciji. Zanimajo nas vse objave ne glede na avtorja in lokacijo, zato se bomo v našem primeru osredotočili na sledenje objavam glede na ključne besede. POST-zahteva, s katero bomo spremljali objave s ključno besedo "bitcoin", bo:

```
https://stream.twitter.com/1.1/statuses
/filter.json?track=bitcoin
```

4.1.2 Spletne novice

Naslednji vir informacij, ki ga bomo spremljali, je vir novic (news feed). To je protokol, preko katerega ponudnik posreduje novice naročnikom. Uporabnik se lahko tako naroči na sprejemanje novic na določenem portalu in ko se seznam objavljenih novic na njem posodobi, uporabnik avtomatično sprejme novo objavljene novice. Uporabnik lahko z uporabo odjemalca (zbiralnika virov) spremlja poljubno število ponudnikov novic.

Vir informacij temelji na XML-formatu, ki vsebuje spletno povezavo do vira novice. Glavna formata, ki se uporabljata za posredovanje novic, sta RSS in Atom [17]. Atom je novejši format in odpravlja nekatere pomanjkljivosti, ki so prisotne v formatu RSS 2.0. Ena od pomembnejših prednosti je modularnost, saj se Atomova sintaksa lahko uporablja tudi v ostalih XML-knjižnicah.

4.1.3 Spletne borze z digitalnimi valutami

Borza z digitalnimi valutami je spletni portal, kjer se na enem mestu srečajo prodajalci ter kupci digitalnih valut (podroben opis je v poglavju 2.3.1).

Vse večje borze digitalnih valut omogočajo dostop do svojih storitev preko brezplačnega programskega vmesnika, ki uporabniku omogoča:

- vpogled v knjigo naročil,
- vpogled v zgodovino transakcij,

- pregled aktivnih naročil in
- izvajanje nakupa/prodaje digitalnih valut.

Za dostop do nekaterih funkcij (npr.: nakup in prodaja digitalnih valut) je potreben ključ programskega vmesnika, ki ga pridobimo z registracijo na izbrani borzi. S tem ključem verificiramo svojo identiteto. Zahteve se pošiljajo preko GET- ali POST-metode, oblika podatkov pa je v JSON-formatu.

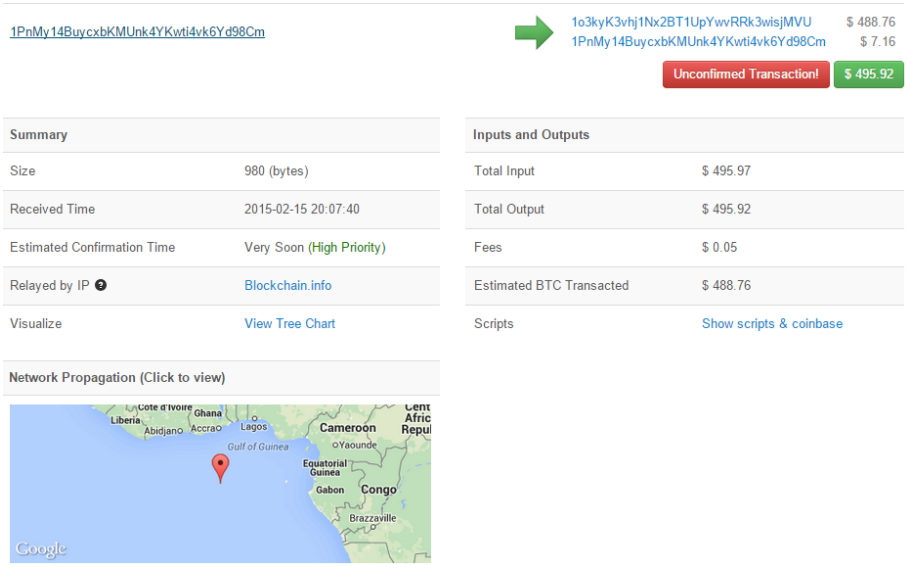
4.1.4 Statistične informacije o delovanju omrežja Bitcoin

Zadnji vir informacij, ki smo ga spremljali, je spletna stran blockchain.info. Osrednja naloga tega spletnega vira je vizualna prezentacija in sledenje vsem transakcijam, ki se izvedejo v omrežju Bitcoin. Primer takega izpisa si lahko ogledamo na Sliki 4.2. Kot vidimo, so vidne vse bistvene podrobnosti o transakciji: naslov pošiljatelja, naslov prejemnika, čas transakcije, celotni znesek, višina provizije ... Poleg omenjenega orodja pa blockchain.info nudi uporabnikom še ostale storitve:

- spletna Bitcoin denarnica,
- podatki o bločni verigi (zgodovini transakcij) in
- ostale statistične informacije (računska zahtevnost, število valute v obtoku, število vseh transakcij, višina provizije s...).

Tudi ta spletna stran omogoča dostop do svojih storitev preko programskega vmesnika, ki je razdeljen na:

- API za procesiranje plačil,
- API, ki se nanaša na uporabo spletne denarnice,
- API, s katerim dostopamo do podrobnih informacijah o transakcijah ter blokih in
- API za dostop do statističnih podatkih o delovanju Bitcoin omrežja.



Slika 4.2: Vizualizacija transakcije

V našem primeru se bomo posluževali predvsem slednjega programskega vmesnika, saj nas zanima delovanje omrežja digitalnih valut kot celote. Menimo, da bi lahko le takšna informacija vplivala na celotno nihanje na trgu. Kljub temu pa ne smemo zanemariti informacije o posameznih transakcijah oziroma blokih, saj se lahko tudi v posamezni transakciji zgodi anomalija, ki bi lahko vplivala na padec oziroma rast vrednosti valute. Kot primer lahko navedemo višino posamezne transakcije; ker ni omejitve o višini transakcije, je lahko obseg ene same transakcije tako velik, da bi ta vplival na spremembo vrednosti valute.

4.2 Podatkovna baza

4.2.1 MongoDB

Za shranjevanje podatkov smo uporabili dokumentno orientirano podatkovno bazo, MongoDB [27]. Ta temelji na nerelacijski NoSQL podatkovni strukturi in ima več prednosti v primerjavi z relacijskimi podatkovnimi bazami.

- Enostavno načrtovanje

Ni potrebno predhodno načrtovanje sheme podatkovne baze. Tako nam ni treba predhodno definirati posameznih zbirk, relacij ter tipov podatkov.

- Horizontalna skalabilnost

  Podatkovna baza je lahko razdeljena med več fizičnimi napravami.

- Fleksibilnost

  Zgradba podatkov se lahko v določeni zbirki poljubno spreminja. To ne povzroča izgube že shranjenih podatkov, prav tako pa ni potrebna rekonstrukcija podatkovnega modela.

- Veliki podatki (big data)

  Podatkovna baza je primerna za spremljanje spletnih virov, kjer je potrebno v relativno kratkem času shraniti veliko količino nestrukturiranih podatkov. Pri spremljanju spletnih virov lahko naš vir spremeni format ter količino podatkov, kar bi predstavljalo veliko težavo v primeru relacijske baze, kjer imamo vnaprej opredeljeno zgradbo podatkovne baze.


```
{
  "high": "255.00",
  "last": "250.00",
  "timestamp": "1422218767",
  "bid": "249.26",
  "vwap": "248.56",
  "volume": "23862.60649405",
  "low": "241.33",
  "ask": "249.99"
}
```


Še ena prednost je format zapisa BSON-podatkov, ki predstavlja nadgradnjo JSON-zapisa. BSON-dokument združuje zbirko večjega števila urejenih elementov, kjer je vsak element sestavljen iz imena, tipa in vrednosti. Ker bomo iz spletnih virov sprejemali odgovore v JSON-formatu (primer zapisa je na Sliki 4.1), to v veliki meri poenostavi rokovanje s podatki. Tako lahko sprejeti JSON-dokument neposredno shranimo v podatkovno bazo brez dodatne rekonstrukcije podatkov.

Poleg enostavnejšega pisanja kode je tu glavna prednost v hitrosti pri zajemu podatkov, ki je ključna pri spremljanju spletnih virov, kjer lahko v zelo kratkem času sprejmemo veliko količino informacij.

4.2.2 Zgradba podatkovne baze

Naša podatkovna baza je sestavljena iz sedmih zbirk:

- bcinfo – statistični podatki, ki jih bomo sprejemali iz spletne strani blockchain.info. Tu shranjujemo celoten JSON odgovor;
- feeds – tu se shranjujejo novice, na katere smo naročeni preko RSS-podajalnika. Shranjujemo časovni žig, naslov ter spletni naslov novice;
- orders – zgodovina knjige naročil. Shranjujemo časovni žig, obseg povpraševanja, obseg ponudb ter vwap (volume weighted average price) za ponudbe in povpraševanja;
- trades – zgodovina tranaskcij. Shranjujemo časovni žig, identifikacijsko številko transakcije, vrednost valute ter količino (znesek) transakcije;
- ticker – kazalci za trenutno vrednost valute. Shranjujemo začatno, končno, najvišjo, najnižjo vrednost ter obseg povpraševanja;
- tweets – sporočila, ki jih sprejemamo iz Twitterjevega programskega vmesnika. Shranjujemo časovni žig, naslov, celotno vsebina sporočila ter označbe, omembe in spletne naslove, ki smo jih izluščili iz sporočila;
- logs – dnevnik napak, ki se ga uporablja za namen razhroščevanja.

4.2.3 Ocena prostorske zahtevnosti

Naziv zbirke	Povpr. velikost posameznega zapisa (Byte)	Perioda shranjevanja (s)	Ocena zasedenosti/dan (kB)
Blockchain	502,6	3600	12,1
News	211,4	spremenljivo	10,5
Orders	122,0	60	175,7
Ticker	176,0	60	253,4
Trades	96,0	spremenljivo	288,0
Tweets	319,0	spremenljivo	446,6

V zgornji tabeli so podani statistični podatki o povprečni velikosti posameznega zapisa, periodi shranjevanja ter ocenjeni prostorski zahtevnosti za vsako od podatkovnih zbirk. Pri zgodovini transakcij, Twitter objavah ter novicah so se objave shranjevale v realnem času, zaradi česar je perioda shranjevanja spremenljiva.

4.3 Implementacija sistema

4.3.1 Twitter

V node.js je bilo spremljanje Twitter objav poenostavljeno z uporabo dveh vtičnikov. Prvi je vtičnik "twit", ki poenostavi povezovanje na Twitterjev programski vmesnik. Drugi pa je "twitter-text", ki poskrbi za sintaktično analizo posameznega sporočila.

V sami inicializaciji povezave na Twitterjev API je potrebno podati štiri ključe:

- consumer key,
- consumer secret,
- access token in
- access token secret.

```
this.t = new twit({
  consumer_key : config.tw.consumerKey,
  consumer_secret : config.tw.consumerSecret,
  access_token : config.tw.tokenKey,
  access_token_secret : config.tw.tokenSecret
});
```

S slednjimi uporabnik identificira sebe ter registrirano aplikacijo, ki bo uporabljala API. Uporabnik mora za pridobitev teh ključev ustvariti račun na Twitterjevi domači strani ter dodati aplikacijo na apps.twitter.com. S tem povežemo našo aplikacijo z našim Twitter računom, ta pa tako pridobi dostop do Twitterjevih storitev.

V naslednjem koraku definiramo tok podatkov, kjer določimo, kateri programski vmesnik bomo uporabili (v našem primeru "statuses/filter"), ter ključne besede, po katerih bomo iskali objave.

```
var stream = this.t.stream('statuses/filter', {
    track : filter
  });
```

V nadaljevanju določimo akcije, ki se izvedejo po prejetem sporočilu, – ko naš tok zasledi novo sporočilo na Twitterjevemu strežniku. V našem primeru bomo izvedli dve akciji. S prvo bomo z uporabo dodatnega filtra izluščili verodostojnejša sporočila, z drugo pa bomo filtrirana sporočila shranili v podatkovno bazo.

Sporočilo, ki ga sprejmemo (data), je zapisano v JSON formatu in vsebuje lastnosti sporočila, ki si jih lahko ogledamo na Sliki 4.1. Glede na lastnosti lahko sporočila dodatno filtriramo. V našem primeru bomo sporočila filtrirali po štirih lastnostih:

- jezik, v katerem je zapisano sporočilo (lang) − zaradi enostavnejše kasnejše analize sentimenta bomo spremljali le sporočila, zapisana v angleškem jeziku;
- pomembnost (filter_level) − filter_level je parameter, s katerim Twitter določa pomembnost sporočila. Trenutno so rezervirane štiri različne stopnje: none, low, medium in high (se še ne uporablja). Pri sporočilih, ki so tako označena

z vrednostjo “medium”, lahko predvidimo, da gre za sporočila z večjo težo. S tem filtrom smo želeli odstraniti vsiljena sporočila (npr. oglasna sporočila);

- ponovno poslana sporočila (retweeted_status) − ta lastnost pove, če je bilo sporočilo že kdaj objavljeno. Retweet je lastnost omrežja Twitter, ki uporabniku omogoča, da ta ponovno objavi sporočilo, katerega avtor je nekdo drug. Ker v našem primeru ne želimo shranjevati podvojenih sporočil, temveč le izvorna sporočila, bomo shranili le sporočila, pri katerih je parameter retweeted_status enak vrednosti “false”;

- popularnost avtorja sporočila (user.followers_count) − ta lastnost pove število sledilcev, ki jih ima avtor sporočila. S tem filtrom smo želeli določiti pomembnost avtorja oziroma njegovo verodostojnost. Predpostavili smo, da so sporočila pomembnejša, če jih je napisal avtor z veliko sledilci. Večje število sledilcev posledično pomeni tudi to, da je to sporočilo prebralo ter delilo naprej večje število uporabnikov.

```
stream.on("tweet", function(data) {
  if (data.lang === 'en' &&
    data.filter_level === 'medium' &&
    !data.retweeted_status &&
    data.user.followers_count > 5000) {
      saveTweet(data);
    }
});
```

V zadnjem koraku kličemo funkcijo, ki shrani filtrirana sporočilo v podatkovno bazo. Namesto celotnega JSON-sporočila se shrani samo za nas pomembne podatke, kot so: čas objave, ime uporabnika, jedro sporočila, omembe ter povezave. V tem modulu (in tudi ostalih modulih) se izvaja beleženje napak pri izvajanju. Če tako pri sprejemanju ali shranjevanju zapisa prihaja do napake, se to ustrezno zapiše v podatkovno bazo, v zbirko “log”.

```
function saveTweet(data) {
  var tweet = {
  "tweet_id" : data.id,
```

```
  "timestamp" : data.timestamp_ms,
  "name" : data.user.name,
  "text" : data.text,
  "hashtags" : t_text.extractHashtags(data.text),
  "mentions" : t_text.extractMentions(data.text),
  "urls" : t_text.extractUrls(data.text)
  };
  col.insert(tweet, function(err, doc) {
    if (err) {
      log.insert({"message" : err,
      timestamp : new Date().getTime()});
    } else {
      log.insert({"message" : "Tweet saved!",
      timestamp : new Date().getTime()});
    }
  });
}
```

4.3.2 RSS

V prvem koraku smo poiskali novičarske portale, ki redno objavljajo novice o valuti Bitcoin ter ostalih digitalnih valutah. Tu smo si pomagali s spletnim mestom "bitcoin.gw.gd", ki že združuje večino spletnih virov, ki objavljajo informacije o digitalnih valutah. S spremljanjem omenjenega spletnega mesta smo izluščili spletne portale, ki pogosteje objavljajo novice o valuti Bitcoin (npr.: CoinDesk [28], Bitcoin magazine [29], newsBTC [30] ...).

V naslednji koraku smo se prijavili na izbrane novičarske portale preko protokola RSS. Tu smo uporabili nodeJS vtičnik "feedsub". Ta podpira naročanje na vire novic, ki uporabljajo protokole RSS, Atom, odgovore pa sprejmemo v JSON-formatu. Modul deluje tako, da ta v določenim intervalu pošilja pogojni GET-zahtevek, ki vsebuje datum zadnje sprejete novice. Na strežniku se nato izvede primerjava z datumom zadnje poslane novice, v odgovor pa se pošlje vse novice, novejše od poslanega datuma.

V primeru, da ponudnik ne podpira GET-zahtevkov, se iz strežnika prične prenos vseh novic, od najnovejše proti starejšim. Vsakič je preverjen tudi datum. Če je ta enak že sprejeti novici, se prenos ustavi, nakar se lahko začne nadaljnja obdelava novic na našem strežniku.

```
function Feed_reader(url) {
  this.reader = new feedSub(url, {
    interval : config.rss.min
  });
  this.reader.on('item', function(item) {
    saveFeed(item);
  });
  this.reader.on('error', function(err) {
    log.insert({"message" : err,
    timestamp : new Date().getTime()});
  });
}
```

Implementacija bralnika novic je prikazana zgoraj. Za osnovno delovanje potrebujemo dva parametra, url naslov podajalnika novic (RSS, Atom ...) ter interval, v katerem bodo poslani zahtevki za preverjanje novic. V asinhronem načinu nato sprejmemo vsako novico posebej, ta pa je v nadaljevanju shranjena v podatkovno bazo. Ker bomo v naslednjem poglavju preverili celotno novico, bomo v tem koraku z razlogom, da prihranimo na prostoru, shranili le naslednje parametre: datum, naslov ter spletni naslov do celotne novice.

4.3.3 Borza digitalnih valut

Za namen te magistrske naloge smo spremljali eno od večjih borz z valuto Bitcoin, Bitstamp. V določenem intervalu smo shranjevali tri različne vrste informacij, knjigo naročil, izvedene transakcije ter vrednost valute.

Pri knjigi naročil shranjujemo časovni žig, ovrednoteno povprečno ceno (Volume-weighted Average Price − VWAP) za ponudbe in povpraševanja ter število naročil in povpraševanj. VWAP nam predstavlja razmerje med vrednostjo valute ter volumnom naročil v določenem časovnem okvirju. VWAP lahko izrazimo z formulo

(4.1), kjer je:

- P – vrednost valute,
- Q – obseg transakcije in
- j – številka posamezne transakcije.

$$P_{VWAP} = \frac{\sum_j P_j.Q_j}{\sum_j Q_j}. \tag{4.1}$$

Pri izvedenih transakcijah shranjujemo poleg časovnega žiga tudi obseg ter ceno, po kateri je bila digitalna valuta prodana.

```
this.exchange.ticker(function(err, candle) {
    if (err) {
      log.insert({"message" : err,
      timestamp : new Date().getTime()});
    } else {
      myself.saveTicker(candle);
    }
  });
Exchange_reader.prototype.saveTicker = function(info) {
  var data = {};
  for ( var el in info) {
    if (info.hasOwnProperty(el)) {
      data[el] = info[el];
    }
  }
  colTicker.insert(data, function(err, doc) {
    if (err) {
      log.insert({"message" : err,
      timestamp : new Date().getTime()});
    } else {
      log.insert({"message" : "Ticker saved!",
       timestamp : new Date().getTime()});
    }
```

```
  });
};
```

Pri trenutni vrednosti valute (ticker) se shranjujejo: začetna, končna, najmanjša, najvišja vrednost valute ter obseg trgovanja. Zgoraj si lahko ogledamo izsek kode, ki skrbi za ažurno shranjevanje vrednosti valute v podatkovno bazo.

4.3.4 Statistični podatki Bitcoin omrežja

Statistične podatke o omrežju Bitcoin bomo zajemali s spletnega mesta blockchain.info. V node.js smo zajem podatkov izvedli enostavno s periodičnim pošiljanjem HTTP-zahtev na spletno mesto "http://blockchain.info/stats?format=json". Celotne odgovore, zapisane v JSON-formatu, nato neposredno shranimo v podatkovno bazo. Spodnji izsek kode predstavlja zahtevek ter povratni klic, v katerem je nato izvedeno shranjevanje podatkovnega niza.

```
function callback(error, response, body) {
  if (!error && response.statusCode == 200) {
    try {
      var info = JSON.parse(body);
      saveBC(info);
    } catch (er) {
      log.insert({"message" : er,
      timestamp : new Date().getTime()});
    }
  } else {
    log.insert({"message" : error,
    timestamp : new Date().getTime()});
    }
  }
request(options, callback);
```

Poglavje 5

Analiza podatkov

V nadaljevanju bodo predstavljene statistične in matematične metode ter končna implementacija, s katero smo želeli določiti primernost posamične časovne vrste pri napovedovanju vrednosti valute.

5.1 Uporabljene metode

5.1.1 Analiza sentimenta

Iz analize javnega mnenja je mogoče z veliko točnostjo napovedati uspešnost nekega produkta na trgu. Primer si lahko ogledamo v delu [7], kjer jim je z veliko točnostjo uspelo iz Twitterjevih objav napovedati uspešnost filmov še pred njihovim izidom. Mi bomo to idejo prenesli na naš primer, kjer želimo ugotoviti, kako vplivajo objave na socialnih omrežjih ter spletnih medijih na nihanje valute.

Takšno analizo imenujemo analiza sentimenta, njen glavni namen pa je ugotoviti razpoloženja iz teksta. V osnovi bomo želeli ugotoviti le polariteto objave, v naprednejši analizi pa bo naš cilj ugotoviti ostala čustvena stanja, kot so jeza, žalost, veselje, negotovost ...

Analizo bomo izvedli z uporabo metode "bag-of-words". Pri tej metodi gre za enostavno razdelitev besedila na posamezne besede, ki se jim nato dodeli vrednost. Vrednost vsake besede je navedena v t. i. slovarju, kjer ima vsaka izrazito pozitivna oziroma negativna beseda navedeno pripadajočo vrednost. Ko se v tekstu ovrednoti vse besede, se te vrednosti sešteje, tako pa se ugotovi polariteta bese-

dila. Slabost tega modela je, da analiziramo le posamezne besede, zaradi česar ne razumemo pomena besednih zvez oziroma zakonitosti, ki jih ima določen jezik.

V naši analizi bomo preizkusili dva različna slovarja:

- AFINN-111 [8] vsebuje 2477 besed in fraz, ki so ročno ovrednostene s strani avtorja [8] z vrednostjo od -5 do 5. Zbirka besed je bila sprva sestavljena iz analize sentimenta Twitter objav, ki so se nanašale na konferenco združenih narodov o podnebnih spremembah (COP15). Ta zbirka se je nato dopolnjevala iz različnih virov – eden od teh je zbirka besed, ki se uporablja v spletnem slengu (http://www.urbandictionary.com);

- Opinion Lexicon [9] vsebuje 6800 besed in fraz, ki so označene le s pozitivno oziroma negativno vrednostjo. Začetna zbirka besed je bila zbrana v delu [9]. Besede so se izluščile iz zbirke recenzij, ki so jih kupci zapisali za različne produkte. Iz vsake rezenzije se je nato poiskalo lastnosti izdelka, ki se pogosto ponavljajo (npr.: objektiv pri fotoaparatu). Ko so bile te lastnosti določene, se je poiskalo pridevnike, vezane na latnost. Polariteto pridevnikov se je nato določilo z pomočjo slovarja WordNet [36].

Kakor vidimo, smo izbrali slovarja, ki se razlikujeta v načinu pridobivanja zbirke besed. Medtem ko je v prvem primeru avtor ročno izbral ter ovrednotil besede, so bile te v drugem primeru izluščene z uporabo algoritma.

5.1.2 Normalizacija

Podatki, ki jih bomo analizirali, bodo predstavljeni v različnih intervalih. Medtem ko je vrednost valute v preteklosti nihala med 0 in 1125 USD, je število dnevnih transakcij doseglo tudi število 100000. Če bi želeli takšni časovni vrsti primerjati, bi bilo podobnost le-teh zelo težko prikazati na grafu. Zaradi enostavnejše vizualizacije ter same analize bomo vrednosti vseh vhodnih podatkov najprej poenotili, tako da bodo te predstavljene v enakih mejnih vrednostih. To bomo storili z uporabo normalizacije, kjer se vse vrednosti v nekem nizu predstavi v meji med 0 in 1 (0 predstavlja največjo, 1 pa najmanjšo vrednost v nizu). Normalizacija se izvede z enačbo (5.1), kjer je:

- X − začetna vrednost,

- X_{min} − najmanjša vrednost v celotnem nizu vrednosti
- X_{max} − največja vrednost v celotnem nizu vrednosti in
- X' − normalizirana vrednost.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{5.1}$$

5.1.3 Združevanje podatkov

Pri nekaterih vrstah podatkov posamičen zapis ne koristi dovolj (npr. obseg posamične transakcije). V podatkovni bazi imamo tako shranjeno vsako posamično transakcijo, kjer je zapisana vrednost ter količina prodane valute. Če bi te transakcije prikazali v časovni vrsti, bi bil graf nepredvidljiv, saj bi prikazovali velikost posamične transakcije namesto obsega trgovanja v nekem časovnem obdobju.

To težavo smo rešili z enostavnim združevanjem podatkov, kjer smo sešteli vrednosti v podanem časovnem okvirju ter to vsoto upoštevali pri izračunu. Primer si lahko ogledamo v spodnjem izseku kode, ki združi polaritete vseh Tweetov, ki so bili objavljeni v okviru ene ure.

```
timeFrame = 3600000;  //1 ura
for ( var i = 0; i < item.length; i++) {
      var txt = item[i].text;
      var time = parseFloat(item[i].timestamp);
      lex.start(txt);
      var marks = lex.res;
      sum += marks;
      if(timeStop<time) {
        arr.push([item[i].timestamp,sum]);
        sum = 0;
        timeStop += timeFrame;
      }
    }
```

5.1.4 Linearna interpolacija

Različne podatke smo zajemali v različnih intervalih. Medtem ko smo vsak Tweet shranili takoj, ko je bil ta objavljen, smo podatke o vrednosti valute shranjevali v minutnih intervalih. Če bi primerjali taki časovni vrsti, ki imata zajete točke na različnih mestih na časovni osi, bi bil izračun korelacijskega faktorja le-teh netočen. Da bomo lahko primerjali dve različni časovni vrsti, bomo morali izračunati (interpolirati) točke v enakih intervalih. To bomo storili z uporabo linearne interpolacije.

Z linearno interpolacijo rešujemo problem, ko želimo določiti vrednost med dvema diskretnima točkama. Vizualno si to lahko predstavljamo tako, da med ti točki narišemo daljico ter na željenem mestu preberemo vrednost. Enačba za linearno interpolacijo [20] je prikazana na (5.2).

$$d = d_1 + \frac{g - g_1}{g_2 - g_1}(d_2 - d_1) \tag{5.2}$$

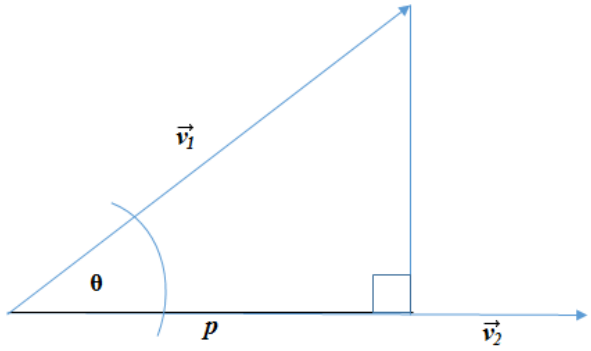
Če uporabimo to formulo pri interpolaciji časovne vrste vrednosti valute, bodo parametri predstavljali:

- g − časovni žig v katerem želimo zvedeti vrednost valute,
- g_1, g_2 − najbližja (mejna) časovna žiga, kjer je vrednost valute znana,
- d_1, d_2 − vrednost valute v mejnima časovnima žigoma in
- d − vrednost valute v iskanem časovnem žigu.

5.1.5 Križna korelacija

Korelacija je številska mera, s katero želimo ugotoviti stopnjo povezanosti dveh časovnih vrst. Ti časovni vrsti oziroma podatkovna vira lahko predstavimo z vektorjema $\vec{v_1}$ in $\vec{v_2}$ (slika 5.1). Kosinus kota θ med njima predstavlja moč povezave dveh podatkovnih virov oziroma stopnjo korelacijskega koeficienta. Ključne vrednosti kota θ so:

- $\theta = 0$: največja pozitivna linearna odvisnost ($\cos(0) = 1$),
- $\theta = \pi$: največja negativna linearna odvisnost ($\cos(\pi) = -1$),
- $\theta = \pi/2$: nekoreliranost ($\cos(\pi/2) = 0$).



Slika 5.1: Grafični prikaz linearne odvisnosti dveh slučajnih spremenljivk

Korelacijski koeficient lahko zajema vrednosti med -1 in 1, linearna odvisnost dveh slučajnih spremenljivk pa je predstavljena z velikim pozitivnim oziroma negativnim korelacijskim koeficientom.

Kosinus kota θ je trigonometrična funkcija, ki je enaka količniku med priležno kateto (p) in hipotenuzo (dolžina vektorja $\vec{v_1}$). Priležno kateto pa izračunamo s pravokotno projekcijo vektorja $\vec{v_1}$ na vektor $\vec{v_2}$. Vektor $\vec{v_1}$ bo v našem primeru predstavljal normalizirano vrednost valute, vektor $\vec{v_2}$ pa normalizirano vrednost opazovanega spletnega vira. Enačba (5.3) prikazuje izračun korelacijskega faktorja iz kosinusa kota θ [18].

$$r = cos(\theta) = \frac{v_1 \cdot v_2}{||v_1|| \quad ||v_2||} = \frac{\sum_i (x(i) - \bar{x})(y(i) - \bar{y})}{\sqrt{\sum_i (x(i) - \bar{x})^2} \sqrt{\sum_i (y(i) - \bar{y})^2}} \tag{5.3}$$

Če pri tem eni od časovnih vrst dodamo časovni zamik t, govorimo o križni korelaciji (5.4).

$$r = \frac{\sum_i (x(i) - \bar{x})(y(i-t) - \bar{y})}{\sqrt{\sum_i (x(i) - \bar{x})^2} \sqrt{\sum_i (y(i-t) - \bar{y})^2}} \tag{5.4}$$

V našem primeru smo primerjali časovno vrsto, ki predstavlja vrednost valute s časovnimi vrstami, ki predstavljajo spletne vire (ponudbe, povpraševanja, število transakcij, polariteta Twitter objav ...). Slednjim smo nato dodali negativen ter

pozitiven časovni zamik, s čimer smo poiskali časovni zamik z največjo stopnjo korelacijskega koeficienta. Pri časovnih vrstah, pri katerih je ob upoštevanju negativnega časovnega zamika ta dovolj velik, lahko predpostaivimo, da le-te vplivajo na vrednost valute. Vseeno pa na tem mestu ne moremo z gotovostjo trditi, da takšna časovna vrsta kljub visoki stopnji korelacijskega koeficienta vpliva na vrednost valute (da ta predstavlja vzrok za nihanje). To bomo dokazali v naslednjem koraku z uporabo regresije ter umetne nevronske mreže.

5.1.6 Studentov t-test

S studentovim t-testom naredimo primerjavo dveh podatkovnih nizov ter ugotovimo, kako se povprečji slednjih razlikujeta med seboj. Z njim želimo zavrniti ničelno domnevo H_0, ki predvideva, da med izmerjenima nizoma ni povezave. Določiti moramo kritično območje $(-\infty, -t_{\alpha/2}) \cup (t_{\alpha/2}, \infty)$, kjer lahko zavržemo ničelno domnevo. Parameter α, s katerim določimo meje kritičnega območja, predstavlja verjetnost za napako 1. vrste oz. napako zavračanja ničelne domneve, ko je le-ta resnična.

Z enačbo (5.5) želimo ugotoviti, če se povprečji nizov razlikujeta med seboj [19]:

- $s_{X_1X_2}$ − standardni odklon,
- $\bar{X}_1$ − vzorčno povprečje iz prvega niza in
- $\bar{X}_2$ − vzorčno povprečje iz drugega niza.

$$t = \frac{\bar{X}_1 - \bar{X}_2}{s_{X_1X_2} \cdot \sqrt{\frac{2}{n}}}; \quad s_{X_1X_2} = \sqrt{\frac{1}{2}(s_{X_1}^2 + s_{X_2}^2)} \tag{5.5}$$

Kako močno opazovani vzorec podpira ničelno domnevo, bomo dokazali z izračunom p-vrednosti. Manjša kot je ta, večji je dokaz za zavrnitev ničelne domneve. Za naš problem bomo za določitev mej kritičnega območja uporabili parameter $\alpha = 0,05$.

Slika 5.2: Grafični vmesnik za izračun stopnje korelacijskega koeficienta

5.2 Implementacija

Primerjava dveh časovnih vrst je bila izvedena z sočasno poizvedbo na ločeni zbirki podatkov, kjer ima časovna vrsta, ki predstavlja spletni vir upoštevan tudi poljuben časovni zamik.

5.2.1 Grafični vmesnik

Na Sliki 5.2 je prikazan grafični vmesnik, ki vizualno prikazuje povezanost dveh časovnih vrst ter izračun stopnje korelacijskega koeficienta. Na vrhu so tri vnosna polja, ker vnesemo začetni in končni datum zajema ter zamik v urah. Na levi strani vmesnika si uporabnik izbere ustrezen tip podatkov, ki jih želi primerjati z nihanjem vrednosti valute Bitcoin. Osrednji del vmesnika predstavlja grafično polje, ki izriše ter primerja časovni vrsti med seboj (modra barva predstavlja valuto, črna pa primerjalno časovno vrsto) glede na izbrane parametre.

5.2.2 Asinhrona funkcija async.js

Node.js se izvaja v sinhronem načinu, zato smo se pri izračunu korelacijskega koeficienta srečali s prvo težavo, saj želimo prebrati podatke iz dveh podatkovnih baz, nato te podatke združiti in na koncu izračunati korelacijski koeficient. Če

bi tako kodo zagnali v sinhronem načinu, bi se najprej izvedla funkcija, ki naredi poizvedbo na prvo bazo, nato pa bi se brez čakanja na odgovor druge funkcije izvedel izračun korelacijskega koeficienta. S tem bi imeli nepopolne podatke in izračun korelacijskega koeficienta bi bil napačen. To težavo smo rešili z uporabo modula async.js, ki okolju node.js doda asinhrono funkcionalnost. Združuje 20 pogosto uporabljenih funkcij, kot so: series (zaporedno izvajanje funkcij), parallel (vzporedno izvajanje funkcij), each (asinhrona zanka) ...

5.2.3 Potek izvajanja

Spodaj si lahko ogledamo osrednjo funkcijo, ki skrbi za izračun in prikaz korelacijkega koeficienta med časovnima vrstama:

```
exports.drawGraph = function(req, res) {
  var type = req.query.type;
  var start = parseFloat(req.query.start);
  var stop = parseFloat(req.query.stop);
  var delay = parseFloat(req.query.delay);
  getCorr(start, stop, type, delay,
  function(callback){
    res.send(callback)
  })
}

function getCorr(start, stop, type, delay, callback) {
  async.parallel([
    function(callback) {
      getPrices(start, stop, callback);
      },
      function(callback) {
    switch (type) {
      case "trades":
        getTrades();
        break;
      case "orders":
```

```
        getOrders();
        break;
      case "asks":
          getOrders(start, stop, delay, "asks_volume",
          min2, max2, callback);
          break;
        case "mining":
          getInfo(start, stop, delay,
          "miners_revenue_btc", min4, max4, callback);
          break;
        case "hash":
          getInfo(start, stop, delay, "hash_rate",
          min5, max5, callback);
          break;
        case "transactions":
          getInfo(start, stop, delay, "n_tx",
          min6, max6, callback);
          break;
        case "trades_info":
          getInfo(start, stop, delay,
          "trade_volume_usd",
          min7, max7, callback);
          break;
        case "tweets":
          getTweets(start, stop, delay,
          min8, max8, callback);
          break;
        case "news":
          getNews(start, stop, delay,
          min8, max8, "link", callback);
          break;
      };
    }
```

```
  ],

  function(err, results) {
    var c1 = results[0]['corr'];
    var c2 = results[1]['corr'];
    var corr = numbers.statistic.correlation(c1, c2);
    results[0]['corr_res'] = corr;
    return callback(results);
  })
};
```

Iz vnosne forme grafičnega vmesnika najprej sprejmemo štiri parametre:

- type − tip podatkov, s katerimi želimo izračunati korelacijski koeficient (orders, asks, trades, Tweets...),
- start − časovni žig začetka analize,
- stop − časovni žig konca analize in
- delay − zamik v urah, opazovane časovne vrste.

V naslednjem koraku se nato pokliče metoda async.parallel, ki sočasno pokliče dve ločeni funkciji. Prva (getPrices) naredi poizvedbo na podatkovno bazo, ki vsebuje zgodovino vrednosti valute Bitcoin. Druga funkcija je odvisna od tipa časovne vrste, ki jo izberemo v vnosni formi. Ta se določi s stikalom "switch", vsaka od njih pa predstavlja poizvedbo po podatkih, s katerimi bomo zgradili primerjalno časovno vrsto.

Ko od obeh funkcij sprejmemo povratni klic, se izvede nova funkcija, ki skrbi za dejanski izračun korelacijskega koeficienta ter pošiljanje podatkov na uporabniški vmesnik. Vsak od povratnih klicev vrne objekt, ki vsebuje dve lastnosti: časovno vrsto, ki jo bomo uporabili za izris grafa ter zaporedje vrednosti, s katerimi bomo izračunali stopnjo korelacijskega koeficienta. Stopnjo korelacijskega koeficienta izračunamo z modulom "numbers" (modul ki združuje več matematičnih in statističnih orodij), rezultat pa dodamo objektu "results", ki ga bomo nato poslali na grafični vmesnik aplikacije.

Glede na tip podatkov se kličejo različne funkcije, kjer vsaka izvaja poizvedbo na posamezno zbirko. Spodaj si na primeru lahko ogledamo funkcijo "getOrders", ki skrbi za poizvedbo na zbirko "orders", kjer so shranjeni informacije o obsegu ponudbe in povpraševanja.

Vsaka funkcija sprejme šest parametrov:

- start − časovni žig začetka izpisa,
- stop − časovni žig konca izpisa,
- delay − zamik časovne vrste v urah,
- min - najmanjša vrednost v časovni vrsti,
- max − največja vrednost v časovni vrsti,
- col - naziv zbirke.

```
function getOrders(start, stop, delay, col,
min, max, callback) {
  delayS = delay * 3600;
  start = start + delayS;
  stop = stop + delayS;
  var objects = {graph:[],corr:[],delay:[]};
  colOrder.find({
    "timestamp" : {
      "$gte" : (start).toString(),
      "$lt" : (stop).toString()
  }
  }, optionsPrice, function(e, item) {
    var arr = [];
    for ( var j = 0; j < item.length; j++) {
      arr.push([parseFloat(item[j].timestamp),
        parseFloat(eval("item[j]."+col))]);
    }
    f = linearInterpolator(arr);
    for(var x = start; x<stop; x+=1000) {
```

```
        var tmp = f(x);
        tmp = (tmp-min)/(max-min);
        objects['corr'].push(tmp);
        objects['graph'].push([(x-delayS)*1000, tmp]);
      }
      callback(null, objects);
    });
}
```

Začetnemu in končnemu časovnemu žigu najprej dodamo časovni zamik (če je ta prisoten). S tem celotno časovno vrsto zamaknemo za določen čas (v urah). V naslednjem koraku naredimo poizvedbo na podatkovno bazo, kjer kot filter uporabimo časovna žiga, kot odgovor pa dobimo vse izpise za ta časovni okvir.

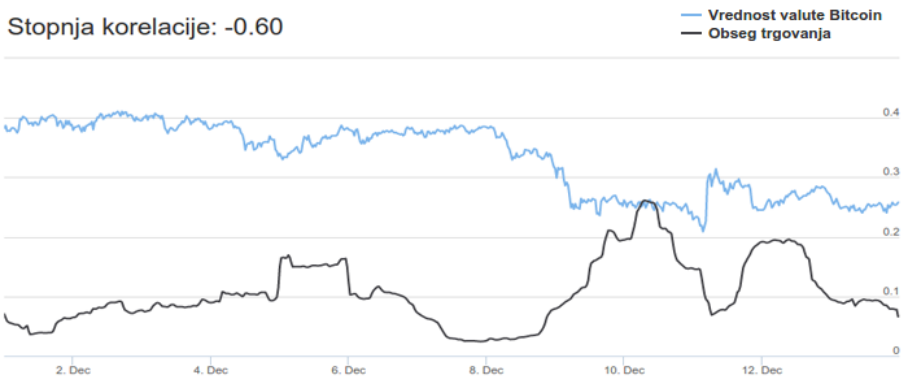
V nadaljevanju preuredimo sprejeti niz podatkov tako, da so ti razporejeni v enakem intervalu. To storimo z linearno interpolacijo. Željene podatke najprej izluščimo iz objekta, ki smo ga sprejeli iz podatkovne baze in jih dodamo v novo polje. Za to polje oziroma zaporedje vrednosti nato izračunamo funkcijo za linearno interpolacijo. S to funkcijo pa nato lahko interpoliramo poljubno točko v našem časovnem okvirju. V našem primeru bomo izračunali točke z enosekundnim razmakom. Tu naredimo le še normalizacijo podatkov ter dodamo vrednost v nov objekt, ki ga vrnemo kot odgovor funkcije.

5.3 Pregled rezultatov

V nadaljevanju bomo naredili pregled izračuna korelacijskega faktorja med vrednostjo valute ter vsemi spletnimi viri, ki smo jih zajeli. Primerjavo smo naredili za 14-dnevni časovni okvir, med 1. 12. 2014 in 15. 12. 2014, kar predstavlja dovolj velik vzorec za točen izračun stopnje korelacije, saj se z večanjem časovnega okvirja ta ni bistveno spremenila.

5.3.1 Zgodovina trgovanja

Za primerjavo zgodovine trgovanja, točneje volumna trgovanja, smo primerjali dve časovni vrsti. Primerjali smo zgodovino, zajeto iz borze Bitstamp ter skupno



Slika 5.3: Zgodovina trgovanja pri zamiku −12 ur

zgodovino trgovanja, ki smo jo zajeli iz spletnega vira Blockchain.info.

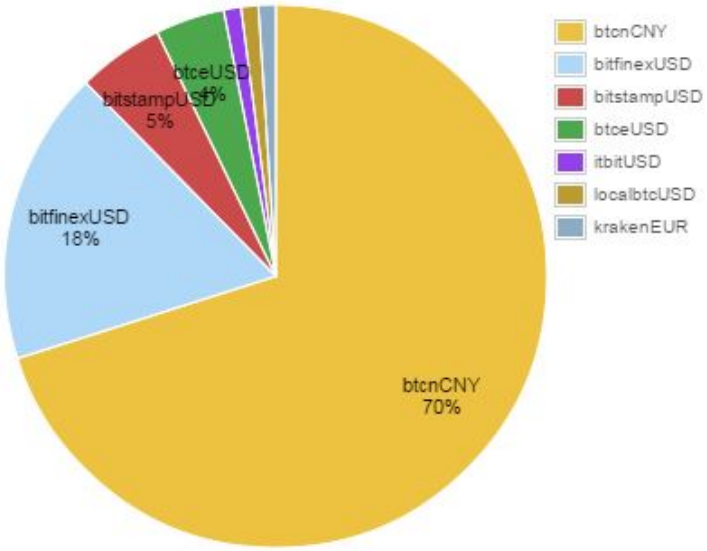
Prišli smo do ugotovitve, da je stopnja korelacije s podatki, zajetimi iz posamične borze, veliko manjša, kot če primerjamo globalno zgodovino transakcij. V prvem primeru je največja stopnja korelacije enaka $-0,23$, medtem ko smo pri zgodovini vseh transakcij dosegli stopnjo korelacije $-0,60$. Razlog za takšno odstopanje je v tem, da v prvem primeru pregledujemo le manjši del vseh trgovanj (Slika 5.4). Kakor vidimo, borza BitStamp zajema le 5 % vseh trgovanj, ki se izvedejo v omrežju Bitcoin, medtem ko močno prevladuje kitajska borza z digitalnimi valutami BTCChina [31] s 70 % deležem. Zaradi omenjene analize relativno majhnega deleža trga je naša časovna vrsta bolj občutljiva na neobičajna trgovanja (npr. posamezno trgovanje z zelo velikim volumnom).

Največjo stopnjo korelacije smo izračunali pri negativnem 12-urnem zamiku (Slika 5.3), iz česar lahko predvidevamo, da zgodovina trgovanja lahko vpliva na nihanje valute. V vseh primerih je bila stopnja korelacije negativna, kar pomeni, da je trend nihanja vrednosti valute nasproten trendu nihanja obsega trgovanja.

Iz zgornjih ugotovitev lahko ustvarimo predpostavko, da manjša aktivnost na trgu digitalnih valut lahko valuti dvigne vrednost in obratno.

5.3.2 Knjiga naročil

Tu smo spremljali dve veličini, obseg povpraševanja ter obseg naročil. V obeh primerih smo pri zamiku 0 ur izračunali najvišjo stopnjo korelacije, 0,79 pri naročilih



Slika 5.4: Porazdelitev obsega trgovanja večjih borz digitalnih valut

(Slika 5.5) ter $-0,61$ pri povpraševanjih (Slika 5.6).

Kot vidimo iz grafov, je glavna razlika med povpraševanji in naročili v predznaku stopnje korelacije, medtem ko imamo pri naročilih visoko stopnjo pozitivne korelacije, je ta pri povpraševanjih negativna. Z večanjem volumna naročil se tako sorazmerno povečuje tudi vrednost valute, medtem ko se z večanjem volumna povpraševanj vrednost valute manjša.

Ker je stopnja korelacije najvišja v primeru, ko ne dodamo časovnega zamika, je težko predvideti, katera časovna vrsta lahko vpliva na drugo – obseg naročil lahko vpliva na nihanje valute ali pa vrednost valute vpliva na obseg povpraševanja. Sta pa kljub temu ta vira močno korelirana z vrednostjo valute, iz česar lahko vidimo močno odvisnost med njima.

Kljub temu, da iz primerjave časovnih vrst ne moremo razbrati, katera vpliva na katero, lahko naredimo naslednjo predpostavko: trgovci (npr.: na trgu vrednostnih papirjev) se večinoma poslužujejo strategije, kjer delnico kupijo, ko je njena vrednost nizka ter prodajo, ko ji vrednost spet naraste. Glede na naše rezultate na področju digitalnih valut veljajo enake zakonitosti, saj vrednost valute vpliva na obseg naročil oziroma povpraševanja in ne obratno. Z upoštevanjem te ugoto-

vitve pa ti časovni vrsti kljub visoki korelaciji ne bosta koristili pri napovedovanju nihanja valute.

5.3.3 Blockchain.info

Primerjali smo tudi vse statistične podatke, ki smo jih zajeli iz spletnega vira blockchain.info. Iz njih smo nato izluščili naslednje časovne vrste, ki so kazale višjo stopnjo korelacije:

- prihodki od rudarjenja,
- računska moč,
- število transakcij in
- skupen obseg trgovanja (že analizirano v prvem podpoglavju).

Na Sliki 5.7 je prikazana časovna vrsta, ki predstavlja provizijo, katero rudarji zaslužijo z rudarjenjem valute Bitcoin. To smo primerjali z časovno vrsto, ki prikazuje vrednost valute Bitcoin ter izračunali linearno odvisnost med njima. Največjo stopnjo korelacije smo izračunali pri časovnem zamiku -8 ur, ko je ta znašala $-0,28$. Do podobnega rezultata smo prišli, ko smo primerjali časovno vrsto, ki prikazuje računsko moč, ki je potrebna za rudarjenje Bitcoin-ov (Slika 5.8). Pri enakem časovnem zamiku smo izračunali najvišjo stopnjo korelacije, ki je enaka $-0,25$. Podobnost provizije z računsko močjo oziroma njuna povezava je razumljiva, saj rudarji pri računanju zgoščevalnih funkcij porabljajo svojo strojne vire (CPU, GPU), za kar so nagrajeni s primerno provizijo.

Bolj presenetljiva je dokaj visoka (negativna) stopnja korelacije teh dveh časovnih nizov z vrednostjo valute. Glede na naše rezultate bi lahko manjša aktivnost rudarjev pri kreiranju novih Bitcoinov vplivala na večjo vrednost valute in obratno.

Naslednja lastnost, ki smo jo primerjali, je število vseh transakcij, ki se izvedejo v omrežju valute Bitcoin. V tem primeru smo izračunali visoko stopnjo pozitivne korelacije. Najvišjo stopnjo korelacije smo izračunali pri zamiku -12 ur, in sicer 0,76. To kaže na to, da bi lahko večja aktivnost na trgu digitalnih valut vplivala na vrednost valute – več transakcij se izvede v omrežju, višja bi lahko bila vrednost valute.

Slika 5.5: Obseg povpraševanja pri zamiku 0 ur

Slika 5.6: Obseg naročil pri zamiku 0 ur

Slika 5.7: Provizija rudarjev pri zamiku -8 ur

Slika 5.8: Računska moč pri zamiku −8 ur

Slika 5.9: Število transakcij pri zamiku −48 ur

5.3.4 Novice in objave na socialnem omrežju Twitter

Zadnji vir, ki smo ga analizirali, je bilo javno mnenje uporabnikov socialnega omrežja Twitter ter spletne novice, ki se nanašajo na valuto Bitcoin.

Iz omrežja Twitter smo zajemali le novice, ki so vsebovale besedo "Bitcoin" in jih nato še dodatno filtrirali (glede na jezik objave, verodostojnost avtorja, število sledilcev ...). V nadaljevanju smo z analizo sentimenta določili polariteto posamezne objave (z oceno od -5 do 5) ter na koncu združili te točke v določenem časovnem okvirju. Slovar, ki smo ga uporabili je AFINN-111 [21], ki smo mu dodali dodali besedne zveze, značilne za omrežje Bitcoin [37]. Končni rezultat smo poizkusili izboljšati tudi z dodatnim upoštevanjem besed iz slovarja Opinion Lexicon [9].

Na podoben način smo analizirali tudi spletne novice. V podatkovno bazo smo shranjevali spletne naslove posamezne novice. Na ta naslov smo se nato povezali ter z razčlenjevanjem izluščili tekst iz opazovane spletne strani. Z enakim postopkom kot pri analizi objav na Twitterju smo nato analizirali polariteto teksta ter ga predstavili na časovni vrsti.

V obeh primerih se je izkazalo, da imata omenjena spletna vira minimalno korelacijo z vrednostjo valute. To lahko vidimo na Sliki 5.10, kjer je prikazana maksimalna stopnja korelacije polaritete objav na Twitterju z vrednostjo valute.

Kakor vidimo, je ta v najboljšem primeru znašala $-0,03$, kar dokazuje, da analizirani časovni vrsti nista povezani.

Rezultati nas ne presenečajo, saj se je tudi primerljiva analiza sentimenta s področja vrednostnih papirjev izkazala za neučinkovito [38]. Objave na socialnih omrežjih pa niso povsem brez teže in vsekakor vsebujejo informacijo, ki bi lahko vplivala na nihanje vrednosti, vendar se tu srečujemo z drugo težavo – kako iz ogromne množice objav izluščiti tiste, ki dejansko vplivajo na vrednost valute.

V članku [38] so tako kritični do rezultatov, ki so objavljeni v [1]. Omenjeno je, da je Twitter le pokazatelj stanja, ki se dogaja v resničnem svetu, in ne orodje za napovedovanje prihodnosti. Analiza sentimenta pa naj bi izhajala iz specifičnih informacij, kot je letno poročilo ter analiza kupcev. Pri analizi vrednostnih papirjev bi bila torej bolj od same novice o določenem podjetju pomembna analiza mnenja uporabnikov o nekem izdelku, ki ga to podjetje trži (npr.: kakšno je mnenje ljudi o novo predstavljenem izdelku).

Slika 5.10: Polariteta objav na socialnem omrežju Twitter pri zamiku −10 ur

Menimo, da bi na podoben način lahko izboljšali tudi rezultat pri analizi trga digitalnih valut. Namesto da analiziramo izključno objave, ki vsebujejo besedo Bitcoin, bi se morali osredotočiti tudi na objave, ki niso neposredno povezane (npr.: objave velikih trgovcev, ki kot možnost plačevanja ponujajo plačevanje z digitalno valuto).

Veliko težavo pa so predstavljale tudi oglaševalske objave (npr.: nagradne igre). Teh nam z našim filtrom ni uspelo izključiti iz analize, saj imajo slednje večinoma veliko število sledilcev, Twitter pa jih vseeno tretira kot verodostojne objave. Takšne objave nimajo nikakršnega vpliva na nihanje vrednosti valute, zato so pripomogle k slabšemu rezultatu same analize.

5.4 Primerjava vzorcev z uporabo studentovega t-testa

V zadnji fazi primerjave smo za potrditev podobnosti nizov izvedli še studentov t-test. Parameter α, ki določa kritično območje in opredeljuje verjetnost za napako 1. vrste, je enak 0,05. Primerjali bomo vrednost valute z vsemi zajetimi spletnimi viri v časovnem obdobju od 1. 12. 2014 do 14. 12. 2014.

Pred samo izvedbo t-testa želimo ugotoviti, če imajo opazovani vzorci normalno porazdelitev podatkov. To smo storili z uporabo Kolmogorov-Smirnovega

testa [39], ki potrdi normalno porazdelitev, če velja $D_n <= D_{n,\alpha}$. Pri določanju kritičnega območja smo upoštevali parameter $\alpha = 5$ %. Za slednjega smo v Kolmogorov-Smirnovi tabeli [40] poiskali kritično vrednost. Nato smo za vsako od časovnih vrst izračunali vrednost Kolmogorov-Smirnov statističnega testa D_n ter primerjali vrednost s kritično vrednostjo $D_{n,\alpha}$. Rezultati so pokazali, da so opazovane časovne vrste približno normalno porazdeljene, saj pri večini časovnih vrst D_n ni presegel kritične vrednosti $D_{n,\alpha}$.

časovna vrsta	časovni zamik ($-t$)	stopnja korelacije	p-vrednost	int. zaupanja (95 %)
Obseg naročil	0	$0,79$	$0,0334$	$-0,0628$ $-0,0026$
Računska moč	8	$-0,25$	$0,0016$	$0,0116$ $0,0511$
Obseg povpraševanj	0	$-0,61$	$3 \cdot 10^{-6}$	$-0,0869$ $-0,0356$
Število transakcij	48	$0,76$	$5 \cdot 10^{-11}$	$0,0673$ $0,1244$
Honorar rudarjev	8	$-0,28$	$0,0006$	$0,0146$ $0,0537$
Obseg trgovanja	12	$-0,60$	$0,0210$	$-0,0984$ $-0,0081$
Polariteta tweetov	10	$0,03$	$0,5120$	$-0,0512$ $0,0256$

V zgornji tabeli so prikazane korelacije, p-vrednosti ter intervali zaupanja za določeno časovno vrsto. Kakor vidimo, je t-test potrdil ničelno domnevo le v enem primeru, pri polariteti tweetov ($p = 0,5120$). To dodatno dokazuje, da ne obstaja povezava med uporabljeno analizo polaritete objav z vrednostjo valute Bitcoin. Pri vseh ostalih časovnih vrstah je vrednost t-statistike ostala znotraj kritičnega območja, s čimer smo zavrgli ničelno domnevo.

Poglavje 6

Simulacija trgovanja

V prejšnjem poglavju smo ugotovili, da imajo naslednje informacije pri določenem negativnem zamiku visoko korelacijo ter so statistično signifikantna:

- obseg trgovanja,
- honorar rudarjev,
- računska moč in
- število vseh transakcij.

Omenjeni podatki predstavljajo kandidate, s katerimi bi bilo mogoče ugotoviti nihanje digitalne valute v prihodnosti. Imamo torej problem, ko želimo poiskati povezavo (realcijo) med večjim številom vhodnih parametrov z enim izhodnim parametrom (vrednost valute). Slednjega lahko rešimo z uporabo dveh metod, ki so se v preteklosti pogosto izkazale za uspešne pri napovedovanju nihanja vrednostnih papirjev ([2] in [3]), večkratno linearno regresijo ter umetno nevronsko mrežo.

6.1 Pregled uporabljenih metod

6.1.1 Večkratna linearna regresija

Z večkratno linearno regresijo želimo modelirati razmerje med več znanimi vhodnimi spremenljivkami (x) z neznano izhodno spremenljivko (y) – kako določen

vhod vpliva na izhod. V našem primeru predstavljajo znane spremenljivke x obseg trgovanja, honorar rudarjev, računsko moč ter število transakcij, neznana spremenljivka y pa predstavlja vrednost valute.

Pri linearni regresiji moramo določiti regresijsko premico, ki najbolj ustreza razmerju med spremenljivkama x in y. To določimo z uporabo opazovanih točk, kjer sta nam znani spremenljivki x in y. Slednja nam pove kako se povprečni odziv spremenljivk x spreminja glede na opazovane točke. Razliko med regresijsko premico ter dejanskimi točkami predstavimo s standardnim odmikom oziroma napako ρ. Vsaka točka odstopa od regresijske premice za odklon ε, velikost slednjega pa je med 0 (točka leži na regresijski premici) in ρ.

Pri metodi najmanjših kvadratov regresijsko premico izračunamo tako, da minimiziramo vsoto kvadratov vertikalnega odklona za vsako opazovano točko. Vrednosti kvadriramo zato, da se negativne in pozitivne vrednosti ne izničujejo.

Enačba za večkratno linearno regresijo za n opazovanih točk, je tako:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip} + \varepsilon_i \quad za \quad i = 1, 2, ..., n \tag{6.1}$$

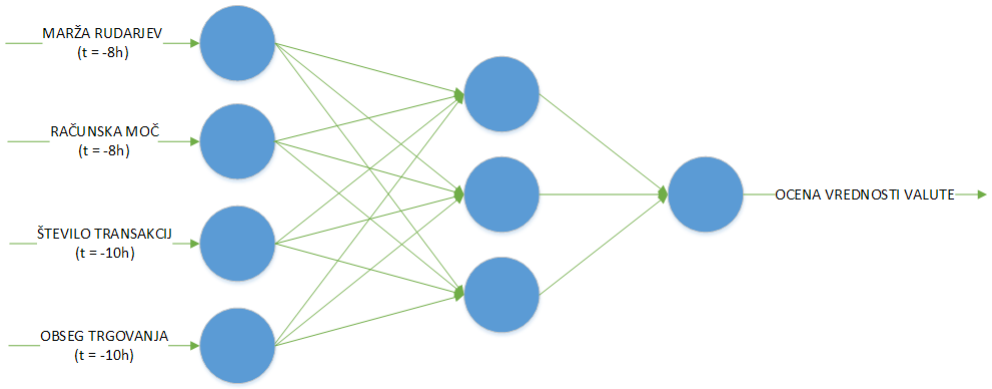
Spremenljivke x bodo v našem primeru predstavljale vhodne parametre (obseg trgovanja, honorar rudarjev, računsko moč ter število transakcij), y pa izhodni parameter oziroma vrednost valute Bitcoin. Z uporabo teh parametrov bomo določili regresijsko premico, katero bomo kasneje uporabili pri napovedovanju vrednosti valute.

6.1.2 Umetna nevronska mreža

Umetna nevronska mreža je učeči se algoritem. Uporablja se ga za določanje približka funkcije, ki preslika večje število vhodov na izhod (funkcija določi izhod glede na znane vhode).

Umetna nevronska mreža je sestavljena iz med seboj povezanih nevronov. Te razdelimo na tri plasti: vhodna, skrita ter izhodna plast. Informacija potuje iz vhodne plasti preko vmesnih v izhodno plast. Povezave med nevroni imajo tudi pripadajočo utež, ta pa predstavlja verjetnost, da bo šla informacija po tej poti (večja je utež, večja je verjetnost, da bo izbrana ta povezava).

Utež se določi v fazi učenja. V tej fazi spustimo skozi sistem niz podatkov, kjer so znani tako vhodi kot izhodi. Tu vsak vhodni niz potuje skozi sistem proti



Slika 6.1: Diagram uporabljene nevronske mreže

pripadajoči izhodni vrednosti. Če se za določeno pot vzpostavi povezava, se uteži na tej poti povečajo.

V naši simulaciji bomo uporabili dva različna modela:

- število skritih nevronov je večje od vhodnih nevronov $(2n + 1)$ in
- število skritih nevronov je manjše od vhodnih nevronov $(n - 1)$.

6.2 Implementacija

Postopek simulacije trgovanja je razdeljen v tri faze:

- učenje,
- napovedovanje in
- simulacija trgovanja.

6.2.1 Uporabljene knjižnice

Jsregress

Jsregress je statistična knjižnica za node.js, ki kot eno od svojih funkcionalnosti ponuja tudi možnost izračuna večkratne linearne regresije.

Pri učenju sistema vhodne parametre definiramo kot zaporedje enačb, ki jih zapišemo v razsežnem polju ($[[eq_1], [eq_2], [eq_3], ..., [eq_N]]$). Vsaka od teh enačb ima obliko $[y, x_1, x_2, x_3, ..., x_M]$, kjer x_M predstavlja vhode, y pa izhode. Te parametre nato dodamo v metodo regression, s čimer izračunamo polinom večkratne linearne regresije, kjer je:

- x_{i1}: marža rudarjev (vhod),
- x_{i2}: računska moč (vhod),
- x_{i3}: število transakcij (vhod),
- x_{i4}: obseg trgovanja (vhod),
- y_i: vrednost valute (izhod) in
- i: posamezen zapis v podatkovni bazi.

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \beta_4 x_{i4} + \varepsilon_i, \quad i = 1, 2, ...n \tag{6.2}$$

Ko je ta izračunan, se pokliče povratni klic z objektom, ki vsebuje omenjeno parabolo (regressedObj). Z uporabo tega objekta lahko v nadaljevanju izračunamo odziv sistema na poljubne vhodne parametre.

```
stat.regression(inputArr, function(regressedObj) {
  console.log(regressedObj.estimate(estArr);
});
```

FANN

FANN [26] je odprtokodna knjižnica s podporo za večplastno umetno nevronsko mrežo. FANN najprej inicializiramo, kjer definiramo število vhodnih, skritih ter izhodnih nevronov:

```
var net = new fann.standard(4,3,1);
```

V fazi učenja podatke definiramo v naslednji obliki:

```
var data = [
    [[x11, x12, x13, x14 ], [y1]],
     [[x21, x22, x23, x24 ], [y2]],
     [[xn1, xn2, xn3, xn4 ], [yn]],
```

```
  ...
];
```

V vsaki vrstici je podan posamezen zapis z opredeljenimi vhodnim ter izhodnim parametrom:

- x_{i1}: marža rudarjev (vhod),
- x_{i2}: računska moč (vhod),
- x_{i3}: število transakcij (vhod),
- x_{i4}: obseg trgovanja (vhod) in
- y_i: vrednost valute (izhod).

Posamičen zapis vhodnih podatkov je tu predstavljeni z x_{i1} (marža rudarjev), x_{i2} (računska moč), x_{i3} (število transakcij) in x_{i4} (obseg trgovanja).

Učenje sistema nato pričnemo s klicem metode net.train(data, error: 0.00001), kjer "error" predstavlja najmanjšo napako, pri kateri bo sistem prenehal z učenjem. Če sistem ne doseže željene minimalne napake, se ta ustavi po sto tisoči iteraciji.

Ko zaključimo z učenjem modela, lahko izračunamo odziv sistema glede na poljuben vhod z metodo: net.run([x1, x2, x3, x4])).

6.2.2 Učenje

Učenje sistema je faza, ko v sistem podajamo tako vhodne kot tudi izhodne parametre. Namen te faze je, da zgradimo model, ki se bo čim bolj natančno odzival na podane vhodne parametre.

Najprej smo izbrali vhodne in izhodne parametre. Vhodni parametri predstavljajo spletne vire, na katerih smo v prejšnjem poglavju izračunali visoko korelacijo z negativnim časovnim zamikom – kandidati, ki bi lahko vplivali na nihanje vrednosti valute v prihodnosti. Izhodni parameter predstavlja vrednost valute Bitcoin. V tej fazi je pomembno, da si izberemo dovolj veliko množico podatkov – večja kot je ta, bolj natančen bo odziv sistema. Za namen te naloge bomo uporabili enomesečne podatke (od 1. 11. 2014 do 30. 11. 2014).

Podobno kot pri izračunu korelacije, smo tudi tokrat najprej poklicali vzporedno asinhrono metodo "async.parallel", ki sočasno naredi več poizvedb na različne

Slika 6.2: Primerjava dejanske vrednosti valute z napovedano vrednostjo pri uporabi večkratne linearne regresije

zbirke ter počaka na odgovore, preden ta nadaljuje z izvajanjem kode. Prav tako poskrbimo za enotno obliko podatkov z uporabo linearne interpolacije ter normalizacije.

Tako urejene podatke nato podajamo v posamezen sistem ter zgradimo model, s katerim bo mogoče glede na vhodne prametre izračunati približek nihanja vrednosti valute. Pri večkratni linearni regresiji kličemo metodo stat.regression(inputData, function(regressionObj) ...), kjer objekt "regressionObj" predstavlja polinom, s katero lahko kasneje izračunamo približek vrednosti valute. Na podoben način pri uporabi umetne nevronske mreže kličemo metodo net.train(inputData, error: 0.00001). V naših simulaciji smo dosegli najmanjšo stopnjo napake v višini 0,00004.

6.2.3 Napovedovanje

V tem koraku poskušamo z uporabo vhodnih parametrov ter naučenega sistema napovedati, kakšno bo nihanje valute v prihodnosti. Na izhodu ne želimo pridobiti povsem točne vrednosti valute, temveč se bomo zadovoljili s kazalci, ki bodo prikazovali mesto višanja oziroma nižanja vrednosti valute.

Podobno kot v fazi učenja tudi v tem koraku naredimo poizvedbe po zbirkah (po izbranih spletnih virih ter vrednosti valute). Bomo pa tokrat uporabili drug časovni okvir kot v fazi učenja, saj bomo le tako dokazali uspešnost predlagane metode (uporabljeni bodo različni časovni okvirji v decembru 2014). Prav tako tokrat v sistem ne bomo podali vrednosti valute, temveč le vhodne parametre

Slika 6.3: Primerjava dejanske vrednosti valute z napovedano vrednostjo pri uporabi umetne nevronske mreže

(spletne vire z negativnim časovnim zamikom). Vrednost valute tokrat služi le kot primerjalna vrednost, s katero bomo primerjali našo napoved.

Tudi tokrat naredimo interpolacijo ter normalizacijo podatkov, nato pa uporabimo modela iz prejšnjega koraka pri napovedovanju. Pri večkratni linearni regresiji to storimo z ukazom "regressedEstimate(inputData)", pri umetni nevronski mreži pa z ukazom "net.run(inputData)".

6.2.4 Simulacija trgovanja

V tej fazi želimo glede na izračunan izhodni niz iz prejšnjega koraka določiti signale za nakup oziroma prodajo valute. Signale za nakup oziroma prodajo valute Bitcoin predstavljajo lokalni maksimumi ter minimumi na grafu – lokalni minimum predstavlja signal za nakup, medtem ko lokalni maksimum predstavlja signal za prodajo.

Na začetku smo določili začetno stanje v USD. V času, ki predstavlja začetek trgovanja, nato kupimo definirano število valute Bitcoin po tečaju, ki velja v tem času. Prodaja oziroma nakup se po tem koraku izvajata izmenično. Po nakupu Bitcoina, začnemo iskati spremembo v časovni vrsti, ko prične vrednost padati. Če padanje preseže prag vrednosti (ta ne predstavlja šuma, temveč dejansko padanje vrednosti), je to pokazatelj, da smo dosegli maksimum, s tem pa signal za prodajo. Enak je postopek pri nakupu – ko v časovni vrsti zaznamo rast vrednosti, je

to signal za nakup. V zaključku trgovanja spet pretvorimo vso preostalo valuto Bitcoin v USD (po trenutnem tečaju) ter to primerjamo z začetnim stanjem – če je končno stanje večje od začetnega, smo dosegli dobiček oziroma v nasprotnem primeru izgubo.

Pri trgovanju smo upoštevali naslednje parametre:

- začetno stanje: 10.000 USD,
- višina transakcije pri nakupu oziroma prodaji: 10 BTC in
- višina provizije: $0,5$ % (po trenutnem Bitstamp ceniku [33]).

Simulacijo smo izvajali 10 dni, od 5. 12. 2014 do 15. 12. 2014. Simulacijo smo naredili za različne vhodne parametre ter kombinacije le-teh:

- honorar rudarjev (X_1),
- računska moč (X_2),
- število vseh transakcij (X_3),
- obseg trgovanja (X_4),
- računska moč + št. transakcij ($X_3 + X_4$) in
- skupaj ($X_1 + X_2 + X_3 + X_4$)

Simulacijo trgovanja smo naredili tako za linearno regresijo kot tudi za umetno nevronsko mrežo. Pri umetni nevronski mreži smo uporabili model z eno vhodno, eno skrito ter eno izhodno plastjo. Pri izbiri števila skritih nevronov smo uporabili dva modela $2n + 1$ ter $n - 1$, kjer n predstavlja število vhodov.

6.3 Pregled rezultatov

Simulacije 10-dnevnih trgovanj z uporabo metod, opisanih v prejšnjem poglavju, so prikazani v spodnji tabeli. V vseh primerih smo določili fiksno začetno stanje (10.000 USD) ter količino Bitcoinov v primeru nakupa/prodaje (10 BTC). V tabeli je za vsako kombinacijo vhodnih podatkov ter napovedovalnega modela prikazan procent dobička oziroma izgube po zaključku simulacije.

	X_1	X_2	X_3	X_4	$X_3 + X_4$	X_1+X_2+ $X_3 + X_4$
Večkratna linearna regresija	$-6,5$ %	$-6,8$ %	$1,2$ %	$0,7$ %	$-0,4$ %	$5,2$ %
Umetna nevronska mreža $(2n+1)$	$-7,9$ %	$-8,1$ %	$-2,7$ %	$1,8$ %	$2,3$ %	$-6,4$ %
Umetna nevronska mreža $(n-1)$	/	/	/	/	/	$-6,2$ %

Pri uporabi večkratne linearne regresije smo imeli največ uspeha, ko smo uporabili kombinacijo vseh štirih vhodnih parametrov. Dosegli smo dobiček v višini 5 %. Ta je bil manjši, ko smo v sistem podajali posamično, število transakcij ter obseg trgovanja, 1,2 % ter 0,7 %. Zanimiva ugotovitev je tudi ta, da smo ob združitvi slednjih, ki sta sicer ustvarjala dobiček, pridelali izgubo ($-0,4$ %). Če smo na vhod podajali le informacijo o računski moči oziroma honorarju rudarjev, smo na izhodu zabeležili večjo izgubo ($-6,5$ % ter $-6,8$ %).

Manj uspešno je bilo trgovanje z uporabo umetne nevronske mreže. Podobno kot pri večkratni linearni regresiji smo tudi tokrat najslabše rezultate dosegli, ko smo na vhod podajali le informacije o porabljeni računski moči ter višini honorarjev. Bistvena sprememba se je zgodila, ko smo uporabili vse štiri vhode – namesto pričakovanega dobička smo prišli do dokaj visoke izgube ($-6,4$ %). Prav tako smo prišli do drugačnega rezultata, ko smo združili število transakcij ter obseg trgovanja (2,3 % dobiček). Rezultate smo poskusili izboljšati s spremembo števila skritih nevronov. Tako smo uporabili model (5,3,1) (Slika 6.1), kjer je število skritih nevronov manjše od nevronov na vhodu sistema. Izboljšava je bila minimalna, saj smo po zaključku simulacije še vedno zabeležili izgubo $-6,2$ %.

Poglavje 7

Zaključek

V tej magistrski nalogi smo želeli z uporabo temeljne analize dokazati, da je mogoče z uporabo spletnih virov napovedati nihanje valute Bitcoin za krajše časovno obdobje v prihodnosti. V raziskavo smo združili veliko skupino informacij, ki so prosto dostopne na spletu. Slednje smo zajemali iz spletne borze digitalnih valut (Bitstamp), socialnega omrežja (Twitter), spletnih novic ter iz spletne storitve, ki beleži statistiko delovanja omrežja Bitcoin (blockcahin.info).

Na začetku nismo imeli informacije, kakšna je povezava posameznega vira z vrednostjo valute, zato smo najprej naredili analizo dotičnih virov. To smo storili z enostavno primerjavo spletnih virov z vrednostjo valute. Uporabili smo križno korelacijo, kjer smo z višanjem negativnega časovnega zamika iskali največjo stopnjo korelacije. S tem pa smo ugotovili, kateri viri bi lahko vplivali na vrednost valute Bitcoin. Tako smo prišli do ugotovitve, da imajo spletni viri, kot so socialna omrežja ter spletne novice, zelo majhen vpliv na opazovano valuto. Po drugi strani smo tudi ugotovili, da imajo nekateri viri kljub visoki stopnji korelacije majhno zmožnost napovedovanja nihanja (ponudba in povpraševanje). Smo pa v tej fazi našli spletne vire, ki so predstavljali možnost za pozitivni izid pri vključitvi leteh v simulacijo trgovanja v naslednjem koraku (računska moč, višina honorarja rudarjev, število transakcij ter obseg trgovanja).

V osrednji fazi simulacije trgovanja smo izvedli simulacijo z uporabo dveh različnih pristopov, večkratne linearne regresije ter umetne nevronske mreže. Tu smo prišli do zanimivih rezultatov, saj se je pokazalo, da je uporaba večkratne linearne regresije veliko uspešnejša v primerjavi z umetno nevronsko mrežo. Do-

kazali smo tudi, da je metoda najuspešnejša, če na vhod podajamo večje število podatkov, za katere smo v fazi analize izračunali visoko stopnjo korelacije.

V naši raziskavi smo dokazali, da je z manjšo točnostjo mogoče napovedati nihanje valute s podatki, ki so prosto dostopni na spletu. Menimo pa, da je še vedno veliko prostora za izboljšave.

Prvi predlog je prilagoditev analize sentimenta pri analizi teksta iz objav na socialnem omrežju. Ugotovili smo, da so zajete objave vsebovale veliko količino šuma (reklamne objave), ki nimajo povezave z vrednostjo valute. S predstavitvijo naprednejšega filtra, ki bi odstranil takšne objave, ter z naprednejšo analizo sentimenta, bi se lahko vključilo tudi omenjene kazalce pri analizi trga digitalnih valut.

Naslednjo predlagano izboljšavo bi dodali v fazi simulacije. Ugotovilo se je, da tudi tu prihaja do nihanj, ki predstavljajo (napačne) signale za nakup oziroma prodajo valute. To pomanjkljivost bi lahko odpravili z uporabo metod, ki so bolj znane s področja tehnične analize. Ena taka možnost zgladitve grafa je uporaba eksponentnega premikajočega povprečja (EMA), s katerim bi zagotovili bolj točne signale pri trgovanju z valuto.

Literatura

[1] J. Bollen, H. Mao, X. Zeng: "Twitter mood predicts the stock market", *Journal of Computational Science*, št. 2, str. 1–8, 2011.

[2] A. Victor Devadoss, T. Antony Alphonnse Ligori: "Stock Prediction using Artificial Neural Networks", *International Journal of Data Mining Techniques and Applications*, št. 2, str. 283–291, 2013.

[3] S. Ghadakchian, A. Pettersen: "Oil Price Changes and the Oslo Stock Exchange", *Lund University, School of Economics and Management*, Bachelor Thesis, 2011.

[4] H. White, D. Pettenuzzo: "Granger causality, exogeneity, cointegration, and economic policy analysis", *Journal of Econometrics*, št. 178, str. 316–330, 2014.

[5] D. Shah, K. Zhang: "Bayesian regression and Bitcoin", *Massachusetts Institute of Technology*, 2014.

[6] R. Grinberg: "Bitcoin: An Innovative Alternative Digital Currency", *Hastings Science & Technology Law Journal*, št. 4, str. 160, 2011.

[7] S. Asur, B. A. Huberman: "Predicting the Future with Social Media", *Web Intelligence and Intelligent Agent Technology (WI-IAT)*, št. 1, str. 492-499, 2010.

[8] F. Å. Nielsen: "A new ANEW: Evaluation of a word list for sentiment analysis in microblogs", *DTU Informatics, Technical University of Denmark, Lyngby, Denmark*, 2011.

[9] M. Hu, B. Liu: "Mining and Summarizing Customer Reviews ", *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, str. 168-177, 2010.

[10] D. Shah, K. Zhang: "Bayesian regression and Bitcoin", *Laboratory for Information and Decision Systems, Department of EECS, Massachusetts Institute of Technology*, 2014.

[11] H. Choi, H. Varian: "Predicting the Present with Google Trends", *Economic Record*, izv. 88, št. 1, str. 2 – 9, 2009.

[12] R. P. Schumaker, H. Chen: "Textual analysis of stock market prediction using breaking financial news: The AZFin text system", *ACM Transactions on Information Systems*, izv. 27, št. 2, čl. 12, 2009.

[13] A. Nikfarjam, E. Emadzadeh, S. Muthaiyah: "Text mining approaches for stock market prediction", *Computer and Automation Engineering (ICCAE)*, izv. 4, št.1, str. 256 – 260, 2010.

[14] E. Havandi, H. Shavandi, A. Ghanbari: "Integration of genetic fuzzy systems and artificial neural networks for stock price forecasting", *Knowladge-Based Systems*, izv. 23 št. 8, str. 800-808, 2010.

[15] M. E. Peck: "The Cryptoanarchists' answer to cash", *IEEE Spectrum*, izv. 49, št. 6, str. 50-56, 2012.

[16] L. Wang: "Investing when volatility fluctuates", *Lee Kong Chian School of Business*, PhD, 2004.

[17] B. Hammersley: "Developing Feeds with RSS and Atom", *O'Reilly*, 2005.

[18] D. Lyon: "The Discrete Fourier Transform, Part 6: Cross-Correlation", *Journal Of Object Technology*, 2010.

[19] J. A. Rice: "Mathematical Statistics and Data Analysis", *Third Edition, Duxbury Advanced*, 2006.

[20] Linearna interpolacija, Dostopno na: `http://www.eng.fsu.edu/~dommelen/courses/eml3100/aids/intpol/` (10. 3. 2015).

[21] Domača stran sentimentalnega slovarja AFINN. Dostopno na `http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010` (10. 3. 2015).

[22] Node.js ogrodje Express. Dostopno na: `http://expressjs.com/` (10. 3. 2015).

[23] CSS/HTML ogrodje Bootstrap. Dostopno na `http://getbootstrap.com/` (10. 3. 2015).

[24] Knjižnica za prikaz grafov in diagramov, HighCharts. Dostopno na: `http://www.highcharts.com/` (10. 3. 2015).

[25] Asinhrona knjižnica za node.js, async.js. Dostopno na: `https://github.com/caolan/async/` (10. 3. 2015).

[26] Knjižnica za umetno nevronsko mrežo, FANN. Dostopno na: `http://leenissen.dk/fann/wp/` [datum dostopa: (10. 3. 2015).

[27] Podatkovna baza MongoDB. Dostopno na: `http://www.mongodb.org/` (10. 3. 2015).

[28] Novičarski portal o digitalnih valutah, CoinDesk. Dostopno na: `http://www.coindesk.com/` (10. 3. 2015).

[29] Novičarski portal o digitalnih valutah, Bitcoin magazine. Dostopno na: `https://bitcoinmagazine.com/` (10. 3. 2015).

[30] Novičarski portal o digitalnih valutah, News BTC. Dostopno na: `http://www.newsbtc.com/` (10. 3. 2015).

[31] Spletna borza BTC China. Dostopno na: `https://www.btcchina.com/` (10. 3. 2015).

[32] Spletna borza Bitfinex. Dostopno na: `https://www.bitfinex.com/` (10. 3. 2015).

[33] Spletna borza Bitstamp. Dostopno na: `https://www.bitstamp.net/` (10. 3. 2015).

[34] Spletna borza OKCoin. Dostopno na: https://www.okcoin.com/ (10. 3. 2015).

[35] Spletna borza ter portal z informacijami o delovanju omrežja Bitcoin Dostopno na: https://blockchain.info/ (10. 3. 2015).

[36] Slovar WordNet. Dostopno na: http://wordnet.princeton.edu/ (10. 3. 2015).

[37] Slovar pogosto uporabljenih besed, ki se nanašajo na omrežje Bitcoin. Dostopno na: http://www.coindesk.com/information/bitcoin-glossary/ (10. 3. 2015).

[38] Will Twitter Ever Be Able To Predict The Stock Market? Dostopno na: http://www.buzzfeed.com/tommywilhelm/will-twitter-ever-be-able-to-predict-the-stock-mar (10. 3. 2015).

[39] Kolmogorov-Smirnov test. Dostopno na: http://www.real-statistics.com/tests-normality-and-symmetry/statistical-tests-normality-symmetry/kolmogorov-smirnov-test/ (10. 3. 2015).

[40] Kolmogorov-Smirnova tabela. Dostopno na: http://www.real-statistics.com/statistics-tables/kolmogorov-smirnov-table/ (10. 3. 2015).